A Methodology for Capability-Based Technology Evaluation for Systems-of-Systems

A Thesis
Presented to
The Academic Faculty

by

Patrick Thomas Biltgen

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
May 2007

A Methodology for Capability-Based Technology Evaluation for Systems-of-Systems

Approved by:

Dimitri N. Mavris, Professor
Advisor and Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Daniel P. Schrage, Professor
School of Aerospace Engineering
Georgia Institute of Technology

David Brown
Air Force Research Laboratory

Robert G. Loewy, Professor
School of Aerospace Engineering
Georgia Institute of Technology

Carlee Bishop
Senior Research Engineer
Georgia Tech Research Institute (GTRI)

Date Approved: 21 February 2007

For Doc...

ACKNOWLEDGEMENTS

Throughout the development of this dissertation I have had many companions that made the journey possible. First and foremost, I would like to thank my family for their unwavering support of my work. My wife, Ashley, has been amazing throughout this process and has helped me in every possible way. I can never thank her enough for her patience, understanding, and love. I would also like to thank all of my friends at ASDL including William Engler, Steven Tangen, Tommer Ender, Reid Thomas, Brian German, Janel Nixon, Shuo-Ju Chou, and Hernando Jiménez for their support and advice.

Additionally, I would like to recognize Barlow Blake and Brad Spearing from Ternion Corporation and John Sall from the SAS Institute who generously provided the software used in this work. Their corporate sponsorship has been instrumental in the development of this research, and I thank them personally for cheering me on as I developed this methodology.

Also, I owe much to the United States Air Force for funding my research through the *AFRL Capability Planning Pilot Project to Support the Air Force Integrated Collaborative Environment (AF-ICE)*, Collaborative Project Order FA8650-05-3-9015 and the American Society of Engineering Education (ASEE) and Air Force Office of Scientific Research (AFOSR) who sponsored me through the National Defense Science and Engineering Graduate Fellowship.

Over the past two years I have been fortunate to have the advisement of a truly world class thesis committee. Dr. Daniel Schrage has played a critical role in teaching me how systems engineering is an integral part of design and how systems-of-systems dominate the world around us. His enthusiastic support of me and my career have been instrumental in my success to date.

I would like to thank Mr. David Brown from the Air Force Research Lab, not only for sponsoring my graduate education, but also for his interest in my work and attention to my development. He continually reminded me to focus my energies and scope the problem at hand to something realistic. Without his support and advice, I would not have succeeded in this endeavor.

Since I met Dr. Carlee Bishop three years ago, she has been a trusted confidante and advocate. Her attention to detail, broad knowledge base, and interest in this topic have been a tremendous help in the development of this work. I also appreciate her sense of humor and enthusiasm throughout the thesis development process.

I am honored and privileged to have Dr. Robert Loewy on my thesis committee. In addition to providing guidance based on his years in the field, Dr. Loewy has served as an inspiration to me. I keep as one of my treasured possessions a letter from Dr. Loewy to a 17-year-old version of myself welcoming me to Georgia Tech and reminding me to "strive for excellence at all times." I will continue to use this as a motto and thank Dr. Loewy for being a role model throughout my development as an aerospace engineer.

Finally and most importantly, I would like to thank my committee chairman and advisor Dr. Dimitri Mavris. Over the past nine years, I have learned lessons about engineering, philosophy, customer relations, finances, management, history, and life in general. Dr. Mavris, you have been my professor, advisor, mentor, tormentor, and friend. I dedicate this work to you, because I can think of no adequate way to thank you for everything you've done for me. You have made the journey through Georgia Tech exciting, educational, and fun, and I could not have made it this far without you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AAA	Anti-Aircraft Artillery, Advanced Aircraft Analysis
ABC	American Broadcasting Company
ABCCC	Airborne Battlefield Command and Control Center
ABM/S	Agent-Based Modeling and Simulation
ACSYNT	Aircraft Synthesis
AD	Area Dominance
ADOC	Air Defense Operations Center
AEA	Airborne Electronic Attack
AF2T2EA	Anticipate, Find, Fix, Track, Target, Engage, Assess
AF2T2EA4	Anticipate, Find, Fix, Track, Target, Engage, Assess, Anyone, Anytime, Anywhere
AFB	Air Force Base
AF-ICE	Air Force Integrated Collaboration Environment
AFRL	Air Force Research Laboratory
AFTL	Air Force Task List
AGM	Air-to-Ground Missile
AHFM	Alternate High Frequency Materials
AI	Artificial Intelligence
AIAA	American Institute of Aeronautics and Astronautics
ALCM	Air Launched Cruise Missile
ALSA	Air Land Sea Application
AMRAAM	Advanced Medium Range Air to Air Missile
AMTI	Air Moving Target Indication
ANOVA	Analysis of Variance
ANSI	American National Standards Institute
AOC	Air Operations Center

ARL	Army Research Laboratory
ARIS	Architecture for Information Systems
ASDL	Aerospace Systems Design Laboratory
ASN	Assistant Secretary of the Navy
ATB	Advanced Technology Bomber, Active Time Battle
AT&L	Acquisition, Test, and Logistics
ATO	Air Tasking Order
ATR	Automatic Target Recognition
AV	All View
AWACS	Airborne Warning and Control System
AWSIM	Air Warfare Simulation
BDA	Battle Damage Assessment
BLU	Bomb Live Unit
BM	Battle Management
BRAINN	Basic Regression Analysis for Integrated Neural Networks
C2	Command and Control
C3	Command, Control, Communications
C4	Command, Control, Communications, Computers
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CAOC	Combined Air Operations Center
CAIV	Cost as an Independent Variable
CALCM	Conventional Air Launched Cruise Missile
CAP	Combat Air Patrol
CAV	Common Aero Vehicle
CCD	Central Composite Design
CC&D	Camouflage, Concealment, and Deception
CDF	Cumulative Distribution Function
CENTCOM	Central Command

CEP	Circular Error Probability
CFAM	Combat Forces Assessment Model
CFTI	Capability-Focused Technology Investment
CID	Combat Identification
CIGSS	Common Imagery Ground Surface System
CJCSI	Chairman of the Joint Chiefs of Staff Instruction
CM	Cruise Missile
CNN	Cable News Network
CONOPS	Concept of Operations
CONUS	Continental United States
CPU	Central Processing Unit
CTOL	Conventional Takeoff
CV	Carrier Variant
DARPA	Defense Advanced Research Projects Agency
DCA	Defensive Counter Air
DCGS	Distributed Common Ground Systems
DDT&E	Design Development Test and Evaluation
DEAD	Destruction of Enemy Air Defenses
DIADS	Digital Integrated Air Defense System
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
DOE	Design of Experiments
DOTMLPF	Doctrine, Organization, Training, Materiel, Leadership, Personnel, and Facilities
DSP	Defense Support Program
DTED	Digital Terrain Elevation Data
EAAGLES	Enhanced Air-to-Air and Air-to-Ground Linked Environment Simulation
EADSIM	Extended Air Defense Simulation

EO	Electro-Optical
EWR	Empty Weight Ration
FALCON	Force Application from the Continental US
F2T2EA	Find, Fix, Track, Target, Engage, Assess
FANNGS	Function Approximating Neural Network Generation System
FIRE	FLAMES Interactive Runtime Executable
FLAMES	Flexible Analysis Modeling and Exercise System
FLARE	FLAMES Analysis and Reduction Environment
FLASH	FLAMES Scenario Highlighter
FLOPS	Flight Optimization System
FLTC	Focused Long Term Challenge
FLTSATCOM	Fleet Satellite Communications
FOR	Field of Regard
FORGE	FLAMES Operational Requirements Graphical Editor
FOUO	For Official Use Only
FPI	Fast Probability Integration
FSA	Future Strike Aircraft
FY	Fiscal Year
GA	Genetic Algorithm
GAO	Government Accountability Office
GBU	Guided Bomb Unit
GATER	Global Air Tactics Employment Review
GBU	Guided Bomb Unit
GCC	Gulf Cooperation Council
GIG	Global Information Grid
GMTI	Ground Moving Target Indication
GPS	Global Positioning System
GSTF	Global Strike Task Force
GTOW	Gross Takeoff Weight

GWAPS	Gulf War Air Power Survey
HARM	High Speed Anti-Radiation Missile
HDBT	Hardened, Deeply Buried Target
HRTI	High-Resolution Terrain Information
HTV	Hypersonic Technology Vehicle
IADS	Integrated Air Defense System
IAEA	International Atomic Energy Agency
ICBM	Intercontinental Ballistic Missile
ICD	Initial Capabilities Document
ICET	Integrating Concept Evaluation Tool
IEEE	Institute of Electrical and Electronics Engineers
IFF	Identification Friend or Foe
IGA	Interactive Genetic Algorithm
IMINT	Imagery Intelligence
INCOSE	International Council on Systems Engineering
INS	Inertial Navigation System
IOC	Initial Operational Capability, Intercept Operations Center
IPPD	Integrated Product and Process Development
IR	Infrared
IRMA	Interactive Reconfigurable Matrix of Alternatives
ISR	Intelligence, Surveillance, and Reconnaissance
ITEM	Integrated Theater Engagement Model
JASA	Joint Airborne SIGINT Architecture
JASSM	Joint Air to Surface Standoff Missile
JCAS	Joint Command and Control Attack Simulation
JCD	Joint Capabilities Document
JCIDS	Joint Capabilities Integration and Development System
JCS	Joint Chiefs of Staff
JDAM	Joint Direct Attack Munition

JECEWSI	Joint Electronic Combat Electronic Warfare Simulation
JIMM	Joint Integrated Mission Model
JFACC	Joint Forces Air Component Commander
JFC	Joint Force Commander
JIOC	Joint Information Operations Center
JPDM	Joint Probability Decision Making
JPEG	Joint Photographic Experts Group
JROC	Joint Requirements Oversight Council
JSAF	Joint Semi Automated Forces
JSTARS	Joint Surveillance and Target Attack Radar System
JSBE	Joint Service Battlespace Environment
JSF	Joint Strike Fighter
J-UCAS	Joint Unmanned Combat Air System
KBS	Knowledge-Based System
LADAR	Light Detection and Ranging
LCMCM	Low Cost Mini Cruise Missile
LGB	Laser-Guided Bomb
LHC	Latin Hypercube
LOCAAS	Low-Cost Autonomous Attack System
LOGSIM	Logistics Simulation
LP	Linear Programming
LRS	Long Range Strike
LRSA	Long Range Strike Aircraft
LRSP	Long Range Strike Platform
MA	Morphological Analysis
MDS	Minimum Detectable Signal
M&S	Modeling and Simulation
MADM	Multi-Attribute Decision Making
MANPADS	Man-Portable Air Defense System

MAS	Multi-Agent Systems
MASINT	Measurement and Signals Intelligence
MBC	Model-Based Calibration
MC2A	Multi-Sensor Command and Control Aircraft
MCDM	Multi-Criteria Decision Making
MCS	Monte Carlo Simulation
MDO	Multidisciplinary Design Optimization
MIL AASPEM	Man-In-the-Loop Air-to-Air System Performance Evaluation Model
MIP	Mixed-Integer Programming
MMORPG	Massively Multiplayer Online Role Playing Game
MODM	Multi-Objective Decision Making
MoE	Measure of Effectiveness
MoP	Measure of Performance
MORS	Military Operations Research Society
MP-RTIP	Multi-Platform Radar Technology Insertion Program
MSRR	Modeling and Simulation Resource Repository
MTI	Moving Target Indication
NACA	National Advisory Council on Aeronautics
NASA	National Aeronautics and Space Administration
NASIC	National Air and Space Intelligence Center
NATO	North Atlantic Treaty Organization
NBC	Nuclear, Biological, and Chemical
NCA	National Command Authority
NCO	Network Centric Operations
NCW	Network Centric Warfare
NDIA	National Defense Industrial Association
NEPP	NASA Engine Performance Program
NGA	National Geospatial-Intelligence Agency
NIMA	National Imagery and Mapping Agency

nm	Nautical Miles
NMD	National Missile Defense
NORAD	North American Aerospace Defense
NPSS	Numerical Propulsion System Simulation
NRC	National Research Council
NSS	National Security Strategy
OCA	Offensive Counter Air
OEC	Overall Evaluation Criterion
ONR	Office of Naval Research
OOP	Object Oriented Programming
OPLAN	Operational Plan
OPR	Overall Pressure Ratio
ORD	Operational Requirements Document
OV	Operational View
P_{Find}	Probability of Locating a Target
P_{Kill}	Probability of Kill
PASS	Principles, Actions, Stakeholders, and Systems
PDF	Probability Density Function
PGS	Prompt Global Strike
PIDO	Process Integration and Design Optimization
PISR	Penetrating Intelligence, Surveillance, and Reconnaissance
POL	Petroleum, Oil, and Lubricants
PoSSEM	Probabilistic System of Systems Effectiveness Methodology
POTUS	President of the United States
PPS	Persistent Precision Strike
PR	Air Force Research Laboratory Propulsion Directorate
PVA	Performance Vector of Attributes
QDR	Quadrennial Defense Review
QFD	Quality Function Deployment

QTA	Quantitative Technology Assessment
RAF	Royal Air Force
RAM	Radar Absorptive Materials
RATTLRS	Revolutionary Approach To Time-critical Long Range Strike
RCS	Radar Cross Section
RDT&E	Research, Development, Test, and Evaluation
RFP	Request for Proposal
RGS	Requirements Generation System
RJ	Rivet Joint
ROE	Rules of Engagement
RSE	Response Surface Equation
RSM	Response Surface Methodology
RTS	Real-Time Strategy
S&T	Science and Technology
SAB	Scientific Advisory Board
SAC	Strategic Air Command
SAE	Society of Automotive Engineers
SAM	Surface to Air Missile
SAR	Synthetic Aperture Radar
SBIRS	Space-Based Infrared System
SBR	Space-Based Radar
SDB	Small Diameter Bomb
SDS	Satellite Data System
SEAD	Suppression of Enemy Air Defenses
SFG	Specific Gross Thrust
SFN	Specific Net Thrust
SHF	Super-High Frequency
SI	Systeme International d'Unites
SIGINT	Signals Intelligence

SIMFORCE	Scalable Integration Model for Objective Resource Capability Evaluations
SLCM	Ship-Launched Cruise Missile
SNR	Signal-to-Noise Ratio
SOC	Sector Operations Center
SOCRATES	Simulation-based Object-oriented Capability-focused Real-Time Architecture-centric Technology Evaluation for Systems-of-Systems
SoS	System(s)-of-Systems
SRAM	Short Range Attack Missile
START	Strategic Arms Reduction Treaty
STK	Satellite Tool Kit
STORM	Synthetic Theater Operations Research Model
STOVL	Short Takeoff and Vertical Landing
STSS	Space Tracking and Surveillance System
SV	Systems View
SWARM	System-Wide Assessment and Research Method
SysML	Systems Modeling Language
TACS	Theater Air Control System
TADIL	Tactical Digital Information Links
TBM	Theater Ballistic Missile
TCT	Time Critical Target
TDA	Technology Development Approach
TEL	Transporter/Erector/Launcher
TESS	Tactical Engagement Simulation Software
THPROP	Thermodynamic Properties
TIE	Tool Integration Environment
TIES	Technology Identification Evaluation and Selection
TIF	Technology Impact Forecasting
TIFF	Tagged Image File Format
TLAM	Tomahawk Land Attack Missile

TPRI	Technology Performance Risk Index
TRADOC	Training and Doctrine Command
TSAT	Transformational Satellite Communications Systems
TSFC	Thrust Specific Fuel Consumption
TV	Technical View
UAE	United Arab Emirates
UAV	Unmanned Aerial Vehicle
UCAV	Unmanned Combat Aerial Vehicle
UGM	Underwater Guided Missile
UJTL	Universal Joint Task List
UFO	Ultra-High Frequency Follow On
UGS	Unattended Ground Station
UHF	Ultra-High Frequency
UK	United Kingdom
UML	Unified Modeling Language
UNSCOM	United National Special Commission
USAF	United States Air Force
USCINCCENT	United States Commander-in-Chief Central Command
USD	United States Dollars
USGS	United States Geologic Survey
UTE	Unified Tradeoff Environment
VA	Air Force Laboratory Vehicles Directorate
VHF	Very High Frequency
VV&A	Verification, Validation, and Accreditation
WMD	Weapons of Mass Destruction
WOC	Wing Operations Center
WOPR	War Operation Plan Response

SUMMARY

Post-Cold War military conflicts have highlighted the need for a flexible, agile joint force responsive to emerging crises around the globe. The 2005 Joint Capabilities Integration and Development System (JCIDS) acquisition policy document mandates a shift away from stove-piped threat-based acquisition to a capability-based model focused on the multiple ways and means of achieving an effect. This shift requires a greater emphasis on scenarios, tactics, and operational concepts during the conceptual phase of design and structured processes for technology evaluation to support this transition are lacking.

In this work, a methodology for quantitative technology evaluation for systems-of-systems is defined. Physics-based models of an aircraft system are exercised within a hierarchical, object-oriented constructive simulation to quantify technology potential in the context of a relevant scenario. A major technical challenge to this approach is the lack of resources to support real-time human-in-the-loop tactical decision making and technology analysis. An approach that uses intelligent agents to create a "Meta-General" capable of forecasting strategic and tactical decisions based on technology inputs is used. To demonstrate the synergy between new technologies and tactics, surrogate models are utilized to provide intelligence to individual agents within the framework and develop a set of tactics that appropriately exploit new technologies.

To address the long run-times associated with constructive military simulations, neural network surrogate models are implemented around the forecasting environment to enable rapid trade studies. Probabilistic techniques are used to quantify uncertainty and richly populate the design space with technology-infused alternatives. Since a large amount of data is produced in the analysis of systems-of-systems, dynamic, interactive visualization techniques are leveraged to enable "what-if" games on assumptions, systems, technologies, tactics, and evolving threats.

The methodology developed in this dissertation is applied to a notional Long Range

Strike air vehicle and system architecture in the context of quantitative technology evaluation for the United States Air Force.

CHAPTER I

MOTIVATION

1.1 Introduction: of Guns and Butter

"Every gun that is made, every warship launched, every rocket fired signifies, in the final sense, a theft from those who hunger and are not fed, those who are cold and not clothed."

-Dwight D. Eisenhower

Engineering is the synthesis of creativity, technology, and science to solve practical problems. Often, these problems span domains and timescales, with complexity that changes dynamically with the world around us. As engineering knowledge advances, man has been able to better allocate finite resources to address challenges to his existence.

This challenge dominates thinking in the defense community: how can resources best be allocated to provide for the common defense of national assets and citizens? In 1959, President Eisenhower warned the military acquisition community to avoid feverishly building massive armaments, saying "expenditures demand both balance and perspective in our planning for defense. At every turn, we must weight, judge, and select. Needless duplication of weapons and forces must be avoided" [140]. The process of evaluating future weapon systems on the basis of effectiveness, durability, and cost continues to be a challenge for military planners.

The challenge is further compounded by the global nature of warfare, the ubiquitous presence of information, and the rapid evolution of threats to our national security. According to General Richard Myers, to provide security in an increasingly dangerous world, "the Armed Forces must be able to evaluate challenges, leverage innovation and technology, and act decisively in pursuit of national goals" [306].

In 1949, the United States Department of War was renamed the Department of Defense

(DoD) with the the mission of providing "the military forces needed to ***deter*** war and to protect the security of the United States[1]" [449]. According to Nitze and McCall, "the best deterrent is possession of superior military fighting capabilities coupled with well-thought-through 'use' and 'declaratory' doctrines" [319]. In the 6th Century B.C., Chinese General and military strategist Sun Tzu echoed this philosophy, saying, "the best victory is when the opponent surrenders of its own accord before there are any actual hostilities... it is best to win without fighting" [416]. The discovery of a robust portfolio of enabling technologies that provide advanced capabilities against emerging challenges is the primary focus of this work.

1.1.1 Strategic Challenges and a Revolution in Military Acquisition

On November 9, 1989, a new era in military policy began with the collapse of the Berlin Wall marking the beginning of the end of the Cold War between the Soviet Union and the United States. The dramatic decline of the Soviet war machine in the early 1990's catalyzed a shift in U.S. military posture from monolithic systems designed to deter all-out war to a more agile and mobile force. During the Cold War, redundancies were seen as strategic and effective: similar effects were produced by multiple systems using different Concepts of Operations (CONOPS), and the stark reality that no defense could stop all offensive systems at once was the cornerstone of the policy of deterrence for forty-five years.

While system acquisition during the Cold War was designed to counter a known adversary with predictable doctrine and strategy, the current strategic environment is dominated by uncertainty and driven by opponents that seek to exploit non-traditional weaknesses in U.S. national security. This point is illustrated by the events of September 11, 2001 and the subsequent "Global War on Terror" which has identified the need for a flexible, agile, responsive force against a wide range of constantly changing and uncertain threats.

According to President George W. Bush, because of this shift the focus is now on "how an adversary might fight rather than where and when a war might occur" [80]. As a result, a system-centric strategy must now shift to define generic ***capabilities***: the ways and means

[1]Emphasis added.

and adversary may choose to accomplish an objective.

Also, the concept of "jointness," or the cross-service employment of military power to accomplish strategic objectives, evolved from the lessons of the 1991 Persian Gulf War. Future military capabilities must avoid a "stovepiped" service-centric focus and must integrate to enable network centric warfare.

To facilitate a transition to flexible, joint, capability-based warfighting, Deputy Secretary of Defense Paul Wolfowitz issued a policy memorandum on October 30, 2003 that identified the need for "an acquisition policy environment that fosters efficiency, flexibility, creativity, and innovation" [483]. In response, an acquisition policy called the Joint Capabilities Integration and Development System (JCIDS) was developed to institute "a capabilities-based approach to identifying current and future gaps in our ability to carry out joint warfighting missions and functions" [123]. This process integrates with the front-end of the Defense Acquisition System defined in DoD Directive 5000 [370, 452].

The focus of the JCIDS is on "identifying, assessing, and prioritizing joint military capability needs" early in the acquisition process; however, the processes and methods used to comply with JCIDS are largely ad-hoc and inconsistent across companies and acquisition programs [370]. A key challenge facing the acquisition community is the evaluation of technology-enabled solutions for which there is no database of empirical data. This challenge is compounded by the complex nature of systems-of-systems required to provide joint capabilities.

1.1.2 Technological Superiority is Key

> *"Linear analysis will get you a much-changed caterpillar, but it won't get you a butterfly."*
>
> -Robert L. Hutchings
>
> Former Chairman of the National Intelligence Council

Technology, from the Greek *techne* meaning "craft" and *logia* meaning "ordering" is "the knowledge of the manipulation of nature for human purposes" [58]. Technology is so critical to the development of civilizations that archaeologists and anthropologists divide human prehistory into epochs based on the technology of the period. While technology can be used to improve life or take it, many of the technological advances of the 20th century were shaped by military development. Ever since the first spear-throwing devices were used in 12,000 B.C., *military technology* has radically shaped the fortunes of those who possess it and those who do not.

Over the past thirty years, exploitation of advanced technology has been the cornerstone of U.S. military policy. This policy traces its origins to the late 1970's when Secretary of Defense Harold Brown and Under Secretary of Defense for Research and Engineering William Perry devised the "offset strategy," which sought to offset Soviet numerical superiority through the development of advanced technology in critical areas [326]. According to Lambeth, "by 1982, the USSR was producing some 1,300 new fighters a year... or a squadron a week and a wing a month" [253]. Outnumbered by a ratio of more than three to one in Eastern Europe, U.S. defense planners pursued "force multipliers aimed at denying the Warsaw Pact any advantage from its numerical edge and offensive doctrine" [253].

The policy of technological superiority is further highlighted in the National Security Strategy of the United States: "Innovation within the armed forces will rest on experimentation with new approaches to warfare, strengthening joint operations, exploiting U.S. intelligence advantages, and taking full advantage of science and technology" [80]. Over the past thirty years, the United States has leveraged its advanced technology in several military conflicts with great success.

Throughout history, the concept of *asymmetric warfare*, where "there is a total or extremely strong difference in the opponents' aims, capabilities, courses of action and moral codes," has been of interest to military planners [362]. Under this paradigm, one side is incapable or unwilling to confront an opponent in a conventional manner and relies on attacks that exploit vulnerabilities in the physical and psychological structure of the adversary. Technologies that enable or negate asymmetric warfare are sometimes called "game changers," which refer to the ability of a technology to change the basic rules of warfare. For example, precision munitions have evolved since the 1970's have changed the way effectiveness is measured. Instead of sorties per target, air power effectiveness is now measured in targets per sortie, as shown in Figure 1. Precision munitions also enable operations against targets in urban environments that would have previously been impossible due to concerns about civilian casualties and collateral damage.

In 1995, air theorist John Warden summarized this dramatic shift in the nature of air warfare:

> *"Precision weapons allow the economical destruction of virtually all targets-especially strategic and operational targets that are difficult to move or conceal.* ***They change the nature of war from one of probability to one of certainty.*** *Wars for millennia have been probability events in which each side launched huge quantities of projectiles (and men) at one another in the hope that enough of the projectiles (and men) would kill enough of the other side to induce retreat or surrender. Probability warfare was chancy at best. It was unpredictable, full of surprises, hard to quantify, and governed by accident. Precision weapons have changed all that. In the Gulf War, we knew with certainty that a single weapon would destroy its target. War moved into the predictable."* [475]

The discovery of such disruptive technologies or "game changers" that either enable an asymmetric advantage on the friendly side or negate potential disruptors on the adversarial side is an area of interest to the Department of Defense [94, 67, 374].

In addition to disruptive technologies, *enabling technologies* also provide similar asymmetric advantages. Enablers are defined as those technologies which, when combined with existing systems and tactics, can facilitate new capabilities otherwise impossible. For example, the invention of ironclad warships in the 19th century was catalyzed by the invention of the steam engine. According to Betz, "although ships could have been clad with iron earlier, they would have been clumsy to sail" [58]. Currently, military decision-makers lack techniques to rapidly assess the mission effectiveness of technology-rich systems-of-systems across a broad trade space encompassing multiple concepts, technologies, tactics, missions,

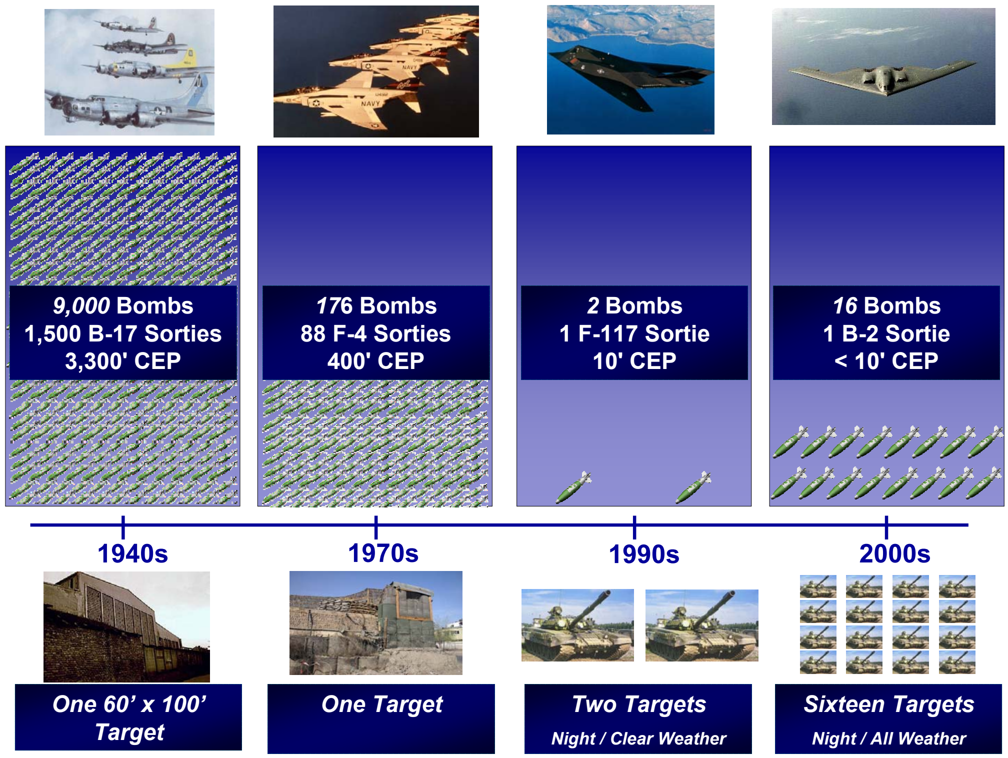


Figure 1: Precision Weapons are a Game Changing Technology.

and domains [36]. In the presence of finite budget limitations, the identification of high-payoff technology areas, game changing technologies, and enabling technologies that provide maximum benefit to the warfighter across these trade spaces is critical.

Finally, Colonel Michael B. Leahy, Director of the Air Vehicles Directorate at the Air Force Research Laboratory identifies the need to "envision alternative futures and then lead the process of discovery, development and transition into war winning solutions." He also notes that "increasingly those solutions can not be achieved by a single technical advance, but rather by integrating a set of science and technology activities toward enabling a system level capability" [256]. The discovery of technologies must therefore be focused on the identification of a portfolio of capability-enabling technologies.

1.1.3 Research Objective

The primary goal for this work is the development of a structured methodology, consistent with the JCIDS, that supports the identification of a robust portfolio of technologies that best provides a capability or capabilities. A key challenge is the integration of multiple heterogeneous elements that comprise a "system-of-systems" that must work together to provide capabilities. The overall research objective is defined below as:

***Research Objective*: The focus of this research is on the development of a valid, defensible, and practical methodology that facilitates a quantitative assessment of technology potential of systems-of-systems with respect to capability-level gaps and provides information to decision-makers early in the design process.**

Exposition of this objective is necessary to clarify the goal and extent of the research described herein. The terms "valid, defensible, and practical" elude to real or perceived shortcomings in current methods for resource allocation. Does the method work? Can the results be defended by the data and analysis? Can the analysis be produced with a reasonable amount of resources? Successfully addressing the balance between these questions will result in a process that is both useful and powerful.

The phrase "quantitative assessment" defines one of the requirements implicit from the previous three terms. A structured methodology that relies on the physics of the problem is a way to quantify the benefit of technologies, but caution is required to stay within the bounds of the practical. The term "technology potential" is defined by Danner as "the possibility for further development or more precisely the availability for further improvement of a technology attribute relative to impending limits" [117]. In this work, technology potential refers not to a forecasted investment-dependent value, but rather the limit of physical realizability of a technology in capability terms.

The ambiguous and oft-used term "capability" is defined in detail in Section 2.1. As an introduction, a capability can be generally defined as “the ability to achieve and effect” and is not tied to a specific physical implementation. The subsequent increase in design freedom is both a curse and a blessing.

Finally, the objective of this research is to cut through the complexity caused by the systems-of-systems nature of JCIDS-compliant acquisitions. The result should be a better quality and quantity of information for judicious use early in the acquisition process.

To formulate a successful approach that uses quantitative analysis to aid decision makers, it is necessary to first review existing methods for technology evaluation to identify whether current best-in-class techniques address the need for capability-based technology evaluation for systems-of-systems.

1.2 Establishing a Baseline: How Technology Evaluation is Done Today

Before proposing the development of a new methodology, it is necessary to review the current state-of-the-art to determine whether an *existing* methodology can be used outright to address the problem at hand. This process is called *baselining*.

According to the Joint Chiefs of Staff, technology evaluation for large-scale heterogeneous system architectures is confounded by two opposing constraints: the need for highly detailed analysis and the desire to maintain the large-scale focus of the problem [372]. This problem is also faced in other fields such as finance, meteorology, software engineering. A review of potential approaches revealed that the ability of an alternative to satisfy requirements is generally determined using qualitative or fuzzy approaches due to the difficulty in modeling complex systems such as weather patterns, financial markets, or commercial user preference.

Within the aerospace community, some popular technology evaluation and resource allocation techniques summarized below include:

- Experimental Approach
- Scientific Advisory Board (SAB)
- Technology Development Approach (TDA)

- Technology Performance Risk Index (TPRI)
- Technology Identification Evaluation and Selection (TIES)
- Quantitative Technology Assessment (QTA)

1.2.1 Experimental Approach

Physical experimentation, the most expensive and least elegant, yet arguably the most effective form of technology evaluation, refers to the direct implementation of a new technology in the field. Extensively used for commercial products such as MP3 players, cellular telephones, and computers, physical experimentation was notably used in the aerospace industry during the 1991 Gulf War. On January 12, 1991, the E-8A Joint STARS aircraft, still in its experimental phase[2] and manned primarily by Northrop Grumman contractors, was rushed into service in the Persian Gulf [348]. Its ability to use Moving Target Indication (MTI) radar detected Iraqi armor moving toward the town of Al-Khafji in what would be the only major Iraqi ground offensive of the Persian Gulf War and allowed U.S. planners to decimate the Iraqi division with air power.

While the experimental approach provides a clear means for assessing capabilities provided by candidate technologies in a realistic operational environment, in the presence of finite fiduciary resources an *analytical* technology evaluation methodology is needed.

1.2.2 Seminar War Games

According to the Department of Defense, a war game is "a simulation, by whatever means, of a military operation involving two or more opposing forces, using rules, data and procedures designed to depict an actual or assumed live situation" [468]. Wargames are typically used to explore new operational concepts, assess alternative force structures, or postulate technology effects using a plausible futuristic scenario. A Seminar War Game is one class of war games that divides a team of experts into two groups that "play" against each other using verbal or written "moves." A game master assesses the results of each move and describes the resulting game board to both teams at the onset of the next move.

[2]Expected to be operational in 1997, but actually became fully operational in late 1998.

Seminar war gaming is used heavily across all services. The Army's School of Advanced Military Studies at Fort Leavenworth, Kansas has conducted war games since the 19th century [335]. The Naval War College instituted the Global War Game, a yearly exercise, in 1978 to extend the Navy's traditional tactical focus to the strategic domain [167]. The Air Force uses two service-operated wargames:

- Global Engagement: "explores emerging operational concepts for employment of air and space power" [432].
- Air Force Future Capabilities Game: "explores alternative futures and force structure to support strategic planning inputs" [432].

These games often take months to set up and execute, require hundreds of people to conduct, and cost millions of dollars [199]. Recent technological advances have extended this basic methodology to include computerized consoles to input game moves; however, few seminar war games utilize advances in computer simulation and gaming to simulate friendly and adversary performance [115].

Seminar War Games also rely almost exclusively on expert judgement. Such exercises may not fully capture the integrated effects brought about by systems-of-systems. Service centric games often lack a focus on joint operations. Finally, these qualitative assessments lack traceability and cannot be extended to off-design conditions.

1.2.3 Scientific Advisory Board

The USAF Scientific Advisory Board[3] (SAB) is an organization tasked with providing long-range forecasting of the research and development needs of the Air Force. Comprised primarily of members from academia, "it provides a link between the Air Force and the nations scientific community" and "promotes the exchange of the latest scientific and technical information that may enhance the accomplishment of the Air Force mission" [435]. The USAF SAB is an advisory body that responds to specific questions posed by senior Air Force leadership and while it may not specifically identify individual technologies, its guidance often focuses Air Force technology policy.

[3] Originally called the Scientific Advisory Group and directed by Theodore von Karman.

Since the formation of the SAB, it has performed six decadal studies on S&T:

- Toward New Horizons (1944)
- Woods Hole Summer Studies (1958)
- Project Forecast (1964)
- New Horizons II (1975)
- Project Forecast II (1986)
- New World Vistas (1996)

Summarizing the SAB reports, Air Force Historian Michael H. Gorn says [181]:

> *"the studies seemed to be entirely random, without connection to one another. They occurred without prior plan; no one organization produced them; their participants varied greatly; their methodologies were not at all uniform; their conclusions varied significantly; and, in fact, they did not even share common purposes."*

While the original purpose of the SAB was to examine advances in science and analyze how these advances may affect the employment of airpower, Gorn notes that over time SAB studies have focused less on physical principles, used more internal Air Force forecasting techniques, and rely less on independent advice [180].

In 1994, the SAB was directed to identify "technologies that will guarantee the air and space superiority of the United States in the 21st Century" [20]. The 15 volume, 2000 page document recommends technologies for specific vehicle classes in support of a number of Air Force missions. The study is very tightly focused on specific vehicles, qualitative information, brainstorming, and anecdotal evidence. In addition to the lack of a structured methodology, the vehicle-centric nature of the later studies makes it difficult to assess the overall effectiveness of proposed technologies *in a systems-of-systems context*. In fact, only the 1964 Project Forecast report filtered candidate technologies with respect to cost, capabilities, and threat assessment [181].

1.2.4 Technology Development Approach

The Technology Development Approach (TDA), developed by Dr. Donald Dix, is a qualitative method for roadmapping expected technology impacts. The TDA (Figure 2) examines several *technology efforts and objectives* and proposes point-estimates for the impacts of each technology. These technologies are then rolled up into the subarea goals (upper right corner of Figure 2) for the proposed system, and extrapolated to expected improvements in top level Measures of Effectiveness (MoEs)[4] in the upper left corner of Figure 2. The TDA is constructed using expert opinion, brainstorming, and qualitative analysis. Shortcomings in this technique include the difficulty of assigning numerical values to expected payoffs, the inability to specify a confidence in the proposed values, a lack of traceability in the analysis process, and the absence of a method to account for the interactions between conflicting or correlated factors.

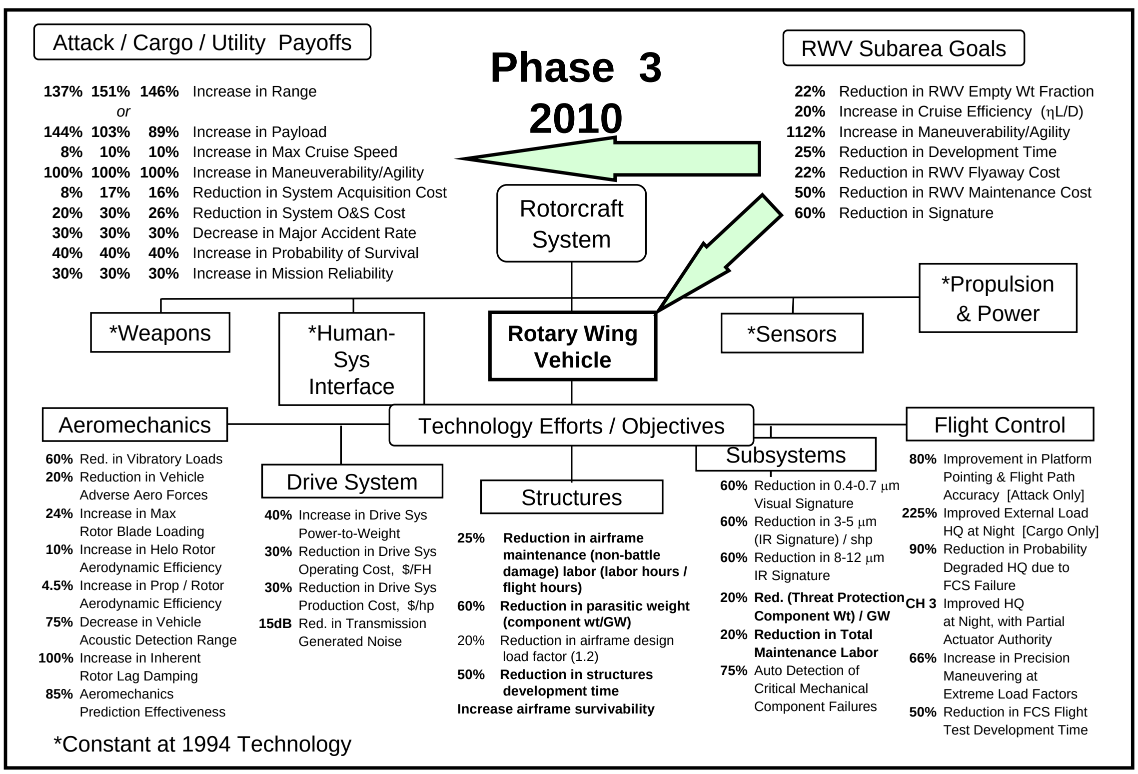


Figure 2: Technology Development Approach for Rotary Wing Vehicles [129].

[4]Section 5.2.2 defines Measures of Effectiveness and Measures of Performance in the context of systems-of-systems.

1.2.5 Technology Performance Risk Index (TPRI)

Noting that the leading cause of problems with weapon system development is immature technology transition, Mahafza proposed an approach called the Technology Performance Risk Index (TPRI) [165]. TPRI is a method for tracking the performance gap and degree of difficulty of a technology throughout its development program [266]. One result of this technique is a measure of the performance achieved relative to the acceptable threshold of performance risk for a given degree of difficulty. For each technology, the TPRI for technology j can be calculated as shown in Equation 1:

$$TPRI_j = \frac{\sum_{i=1}^{n} 1 - \frac{A_i}{1+(1-A_i)DD_i}}{n} \tag{1}$$

Where A_i is the i^{th} Measure of Performance (MoP), DD_i is the degree of difficulty for meeting the required MoP, and n is the number of MoPs being considered. This measure is meant to be exercised at different time levels of technology maturity until the numerical value of TPRI is zero.

While TPRI provides transparency across the technology development life cycle, it is not well-suited for system-of-systems programs as it tracks only MoPs and not MoEs. Furthermore, it is heavily based on qualitative information and lacks a mechanism for accounting for multiple technologies simultaneously.

1.2.6 Technology Identification, Evaluation, and Selection (TIES)

Kirby's Technology Identification, Evaluation and Selection (TIES) methodology is a "comprehensive and structured method to allow for the design of complex systems which result in high quality and competitive cost to meet future, aggressive customer requirements" [239]. This technique uses modeling and simulation to quantitatively assess the impact of technologies by representing the technology impacts as "k-factors" (see Section C.2). While TIES can be seen as a quantitative extension of the TDA approach, traditional applications of the method have been primarily focused on the evaluation of MoPs for a system and have to date not addressed the issue technology evaluation for large-scale heterogeneous systems.

For example, Hale notes that technology evaluation for systems-of-systems require integration of one or more scenarios and involve "significant interactions between the platform and specific missions" [187]. The TIES method also assumes that "the impacts of the individual technologies are additive," which may not be a valid assumption for systems-of-systems that are dominated by nonlinearity [240, 484]. However, elements of the TIES method, most notably the integral focus on modeling and simulation as a means to calculate the performance of a system and the use of "k-factors" to represent technology impacts, provide a framework for enabling quantitative technology evaluation for systems-of-systems.

1.2.7 Quantitative Technology Assessment (QTA)

The United States Air Force Research Laboratory (AFRL) is actively engaged in a research effort to "integrate new methodologies and tools with existing 'industry-standard' tools to effectively test the effects of new technologies on air vehicle capability" [481]. The "approach requires that the Air Force be able to quantify the impacts of any proposed technology on each key capability" [395]. Under this paradigm, a program called Quantitative Technology Assessment (QTA) has been initiated that aims to reduce analytical cycle time, combines multiple disciplinary models, and enables broader design space exploration. According to AFRL program manager David Brown, QTA provides a traceable process that enables informed R&D decisions [76]. In addition to capability gap analysis, the QTA process shown in Figure 3 provides "meaningful mission effectiveness analysis that quantitatively measures the value of technologies" [76]. Under the framework of QTA, an environment can also be constructed that enables real-time interaction between geographically distributed design organization through direct linking of simulation tools as shown in Figure 4.

AFRL Simulation Based Research and Development Lead James Zeh says QTA is enabled through constructive simulation and parametric modeling [495]. Caudill and Zeh also note than a simulation architecture should be modular and flexible to support interchangeability of modules and "limit dependency on system specific models" [90]. Quantitative Technology Assessment is well suited for system-of-system studies and evaluation of technologies with respect to capability-level MoEs.

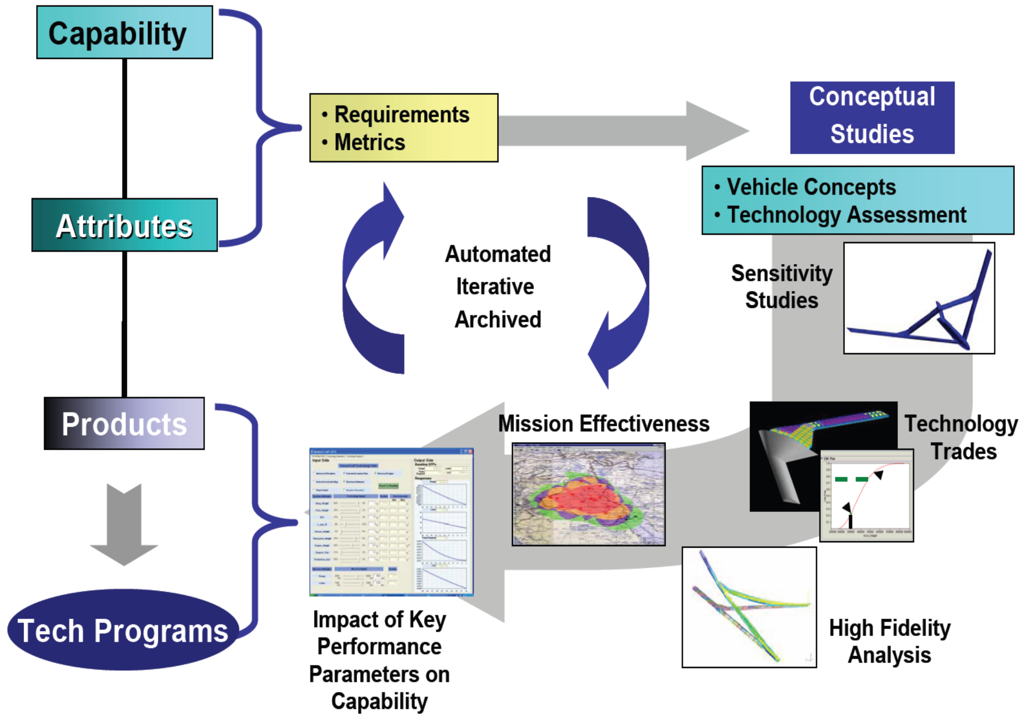


Figure 3: Outline of the QTA Process [76].

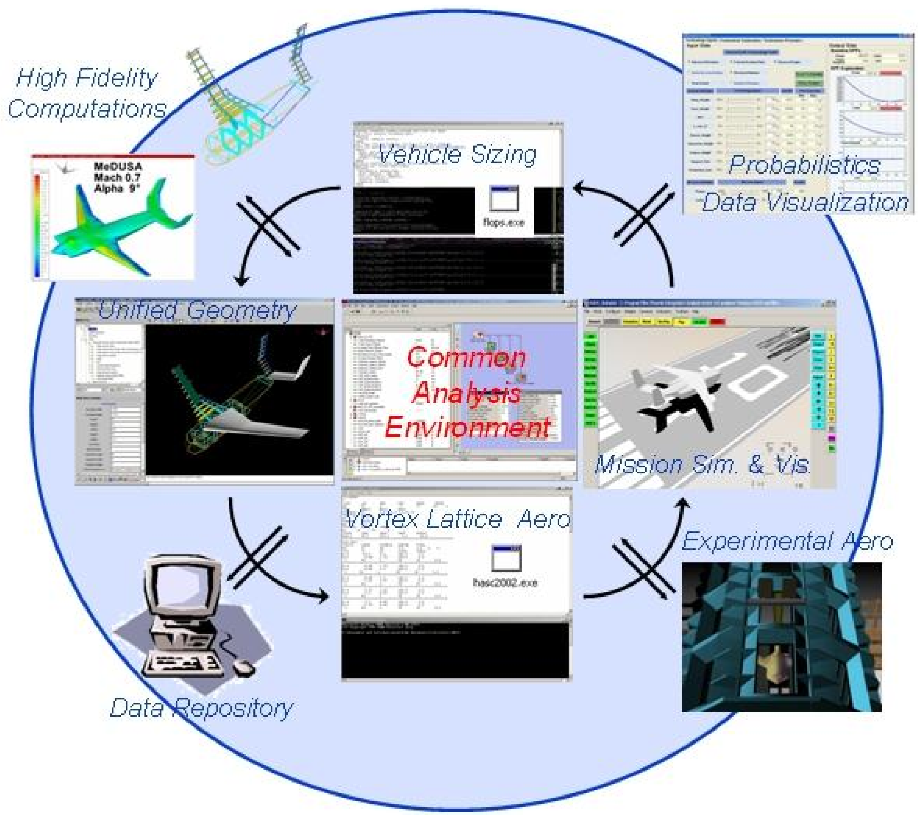


Figure 4: Quantitative Technology Assessment Environment and Links [13].

1.2.8 Summary of Existing Technology Evaluation Methods

After a review of the above techniques, several key attributes of popular technology evaluation methods emerge:

- Quantitative: Measurable process for enumerating ways and means and comparing solutions
- Traceable: Enables identification of the effectiveness drivers of a proposed technology solution.
- Flexible: Generalizable to multiple problems in the same class with minimal modification
- Reusable: Method and environment can be used to study multiple attributes of the same problem.
- Rapid: Can be applied in a reasonable time frame without unrealistic resource requirements
- Parametric: Avoids point solutions and provides visibility into behaviors previously obscured by the complexity of the problem
- Scalable (to Systems-of-Systems): Avoids simplistic representations of interactions between systems
- Affordable: Produces valid results without extensive manpower commitments and uses commercial off-the-shelf tools when possible
- Simple: The steps in the methodology are reasonable, logical and teachable

Based on these definitions, the aforementioned techniques can be qualitatively compared based on the author's assessment across the multiple attributes as shown in Figure 5.

As the figure illustrates, none of the methods surveyed are ranked excellent in all dimensions. The green shaded boxes identify the best-in-class techniques for each of the attributes as identified by the author. All things being equal, the TIES methodology is the best-in-class technique across all identified dimensions; however, QTA is a set of "methods/tools/processes that allow an assessment of the impact/value/contribution of technologies in Air Force capability based terms" [77]. The synthesis of aspects from the TIES and

	Experimental	Seminar War Game	SAB	TDA	TPRI	TIES	QTA
Quantitative	Very Good	Fair	Fair	Fair	Good	Very Good	Excellent (Best-in-Class)
Traceable	Excellent	Poor	Good	Very Good	Very Good	Very Good	Excellent (Best-in-Class)
Flexible	Poor	Very Good	Fair	Very Good	Good	Excellent (Best-in-Class)	Very Good
Reusable	Fair	Fair	Poor	Very Good	Fair	Very Good (Best-in-Class)	Very Good
Rapid	Fair	Good	Fair	Very Good (Best-in-Class)	Very Good	Good	Fair
Parametric	Poor	Poor	Poor	Fair	Fair	Excellent (Best-in-Class)	Very Good
Scalable to SoS	Poor	Good	Fair	Fair	Poor	Good	Excellent (Best-in-Class)
Affordable	Poor	Good	Very Good	Excellent (Best-in-Class)	Excellent	Good	Fair
Simple	Very Good	Very Good	Very Good	Excellent	Excellent (Best-in-Class)	Good	Good
Overall	Fair	Fair	Good	Good	Good	Very Good (Best-in-Class)	Very Good (Best-in-Class)

Excellent | Very Good | Good | Fair | Poor | Best-in-Class

Figure 5: Characteristics of Several Technology Evaluation Methodologies.

QTA methods facilitates the definition of a new method that is well suited for capability-based technology evaluation for systems-of-systems. To meet the increasing analysis needs of the acquisition community, a new method should support a variable-fidelity approach to modeling and simulation. Techniques to speed up the analysis process and enable trade space exploration across a multivariate problem are also needed to increase the usability of results. Furthermore, none of the methods surveyed account for the confounding impact of tactics on technology selection or the difficulty in analyzing disparate architecture technologies. These shortcomings *drive the need for a new methodology* that includes these elements.

1.2.9 A New Methodology is Needed

Since there is no existing technique that completely addresses the needs of the technology evaluation community, the purpose of this research is to extend the basic concepts of the existing processes through infusion of new techniques and methods. The "product" of this research is a new methodology, "a body of practices, procedures, and rules used by those who work in a discipline" [22]. The primary expected payoff in this research is the codification of a structured methodology for technology evaluation that can be extended to address a range of problems in the field of systems-of-systems engineering. Additionally, the demonstration of this process results in the creation of a parametric tradeoff environment for systems and technologies.

The measures of success used to evaluate the effectiveness of the proposed methodology are based on the qualitative attributes identified in Figure 5. These attributes are revisited in the conclusion to assess the ability of the proposed methodology to address shortcomings of existing resource allocation and technology forecasting techniques.

1.3 Defining an Example Application

A new planning framework called "Focused Long Term Challenges" (FLTCs) has recently been implemented to provide a direct link between technology evaluation and capabilities-based planning at the Air Force Research Laboratory (AFRL) [371]. The "FLTCs build upon the Long-Term Challenges identified in the comprehensive S&T Planning Review undertaken several years ago at the direction of Congress and will guide investment" in the AFRL technology portfolio [219].

As of August 2006, the FLTCs are Anticipatory Synchronized Operations, Tailored, Persistent Collection for Predictive Battlespace Awareness, Acquire and Engage Difficult Targets, Assured Operations in High Threat Environments, Integrated Cyber/Info Effects, Responsive Adaptive Theater Operations, and Affordable Aerospace Reliability and Readiness (see Figure 6) [64].

To address the challenges identified in the FLTCs, the Air Force Research Laboratory Vehicles Directorate defines seven supporting capabilities: Cooperative Airspace Operations,

1. **Anticipatory Synchronized Operations**
2. **Tailored, Persistent, Collection for Predictive BattleSpace Awareness**
3. **Acquire & Engage Difficult Targets**
4. **Assured Operations in High Threat Environments**
5. **Integrated Cyber/Info Effects**
6. **Responsive Adaptive Theater Operations**
7. **Affordable Aerospace Reliability and Readiness**

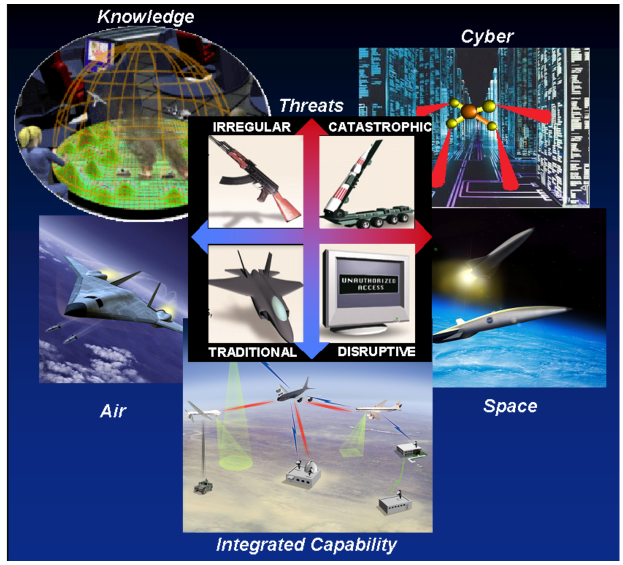


Figure 6: Air Force Focused Long Term Challenges (FLTCs), LRS-relevant Challenges Highlighted in Red [64].

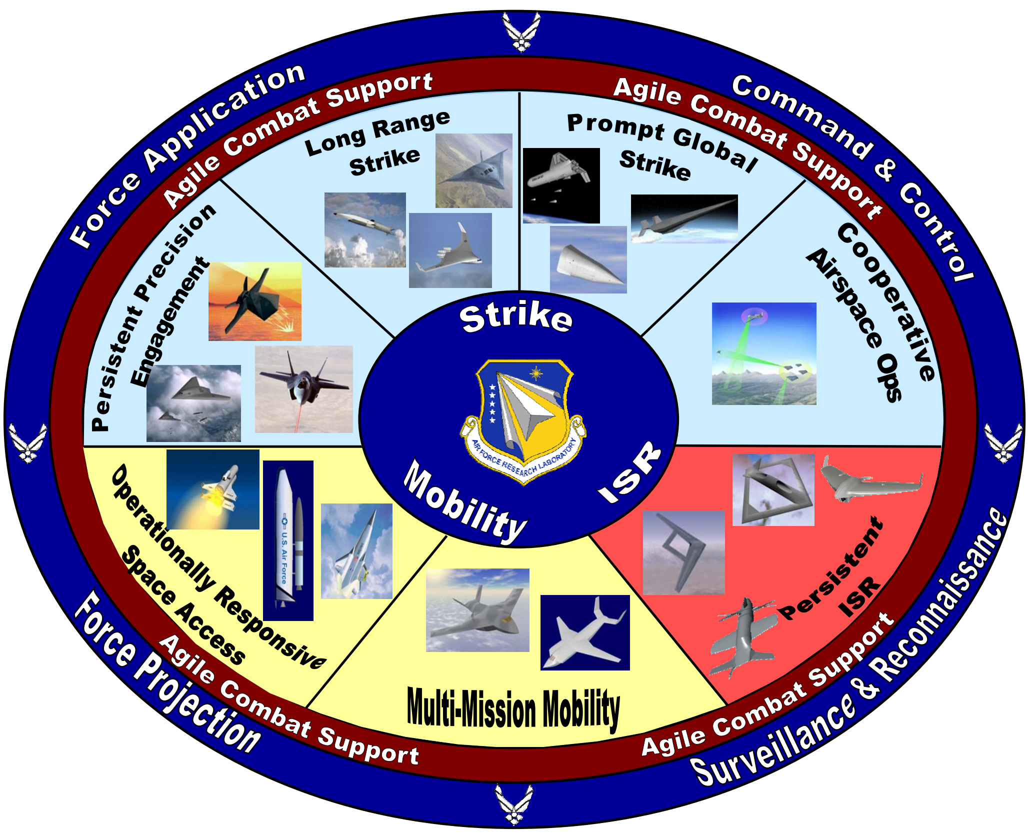


Figure 7: Capabilities Defined by the Air Force Research Lab Vehicles Directorate [444].

Multi-Mission Mobility, Operational Responsive Space Access, Persistent ISR, Precision Persistent Strike, Prompt Global Strike, and Long Range Strike shown in Figure 7 [444]. Of the seven capabilities, Long Range Strike (LRS) has been gaining increased importance Congress, within the aerospace community, and in the press. Congress has noted that evolving LRS capability through an improvement of the nation's bomber force is a high priority [201, 408, 478].

Long Range Strike (LRS) capability can be defined as "the ability to conduct prompt, accurate, conventional strikes anywhere on the globe on very short notice" [477]. The concept can also be defined from an *effects-based* standpoint: "achieve a desired effect(s) rapidly and/or persistently, on any target, in any environment, anywhere, at any time." [444]. The 2006 Air Force Posture Statement notes that "responsive capabilities will combine speed, stealth, and payload to strike hardened, deeply buried, or mobile targets, deep in enemy territory, in adverse weather and with survivable persistence" [305]. Finally, providing increased LRS capability directly addresses FLTCs three and four, highlighted in Figure 6.

Current Air Force Doctrine defines the LRS as an element of the Global Strike Task Force (GSTF) [230]. In this context, the primary objective for LRS assets is "kicking down the door" to open the way for the rest of the U.S. military. According to former Air Force Chief of Staff General John P. Jumper, the GSTF "is not necessarily your war-winning force. It creates the conditions for the war-winning force to get close enough to do their job" [331]. LRS capability is defined in detail in Section 2.4.

Long Range Strike capability impacts a number of the high-level strategic challenges facing the United States through *the ability to quickly degrade enemy offensive capability with particular emphasis on high-payoff elements such as the enemy's weapons of mass destruction development, production, and employment assets.* The relevance of this problem, the heterogeneous nature of LRS architecture elements, and the availability of public domain data related to potential LRS concepts and applications drive the selection of Long Range Strike capability for the demonstration of the proposed methodology.

1.4 Dissertation Organization

In Chapter I, a key issue facing the acquisition community, the lack of a structured methodology for performing quantitative technology evaluation for systems-of-systems was surveyed. This problem motivates the development of a new methodology. Chapter II further elaborates on the problem by defining key terms, expounding upon the example application, and summarizing key assumptions needed to establish an appropriate "control volume" for methodology development. In Chapter III, several technical barriers to the development of a new methodology are revealed, research questions are formulated to address these challenges, and hypotheses are proposed that identify enabling techniques based on each research question. In Chapter IV, a methodology for capability-based technology evaluation for systems-of-systems is proposed. This methodology addresses the shortcomings of existing methods defined in Section 1.2 and is designed to incorporate and synthesize the hypotheses identified in Chapter III.

A key aspect of engineering dissertations is the need to validate the philosophical arguments from Chapters III and IV using an experimental test of some sort. Chapter V details a modeling and simulation environment developed to test the proposed methodology using a Long Range Strike system architecture. Finally, Chapter VI summarizes the lessons learned from the application of the proposed methodology to the Long Range Strike problem and identify areas for future work.

In addition to the body of this dissertation, several appendices that summarize a search of the technical literature are included to justify the proposed hypotheses in Chapter III. The proposed hypotheses are summarized in Section 3.4 and a taxonomy of options is illustrated in the form of a matrix of alternatives in Figure 22. Readers unfamiliar with the options proposed may wish to review the appendices for a more thorough exposition of the individual techniques, tools, and methods synthesized in this work.

CHAPTER II

PROBLEM DEFINITION

Section 1.2.9 identified the need to develop a new methodology due to shortcomings in existing techniques and methods. The first step of the problem definition phase is to establish the characteristics of the new methodology. This includes a definition of relevant terms and an identification of possible challenges related to methodology development. Next, Long Range Strike capability is in Section 1.3 as a challenging problem to test the implementation of the proposed methodology. LRS capability is defined in detail in Section 2.4. Finally, a recapitulation of key observations made throughout this section is given in Section 2.5. These observations shape the research questions and hypotheses defined in the subsequent chapter.

The title of this work is ***A Methodology for Capability-Based Technology Evaluation for Systems-of-Systems***. The scope of work can be best delineated by defining each of these terms in turn.

2.1 Definition of a Capability

Throughout the Cold War, military acquisition was *threat-based*: system procurement was based on anticipated threats posed by a known enemy, the Soviet Union. Since systems have significant development cycle times, requirements were based on forecasted performance of systems the enemy was expected to employ by the entry-into-service date. While this forecasting process was dominated by assumptions and uncertainty, degrees-of-freedom related to the geography, policy, doctrine, and technology readiness were somewhat constrained.

While this strategy was effective, the current military situation is drastically different from that of the past forty years. Nation-states with fixed emplacements, tank columns, airfields, and strategic weapons have given way to smaller, fragmented, non-governmental enemies which generally lack sophisticated integrated defensive systems. For example, during Operation *Enduring Freedom* in 2001, B-2 bombers only operated for two days, by which time a majority of Afghanistan's air defenses were destroyed [183]. Defense Secretary Rumsfeld went on to say that "[The Afghans] do not have high-value targets or assets that are the kinds of things that would lend themselves to substantial damage from the air" [361].

Furthermore, the acquisition policies for traditional weapons systems were driven by service-centric requirements codified in an Operational Requirements Document (ORD). "These documents have tended to spawn rigid acquisition programs that focus too narrowly on achieving specified and sometimes overly optimistic levels of performance" [463]. In the interest of protecting cost and schedule, new technologies that add programmatic risk may be excluded. The result of the ORD-based process was that systems were approaching obsolescence by the time they were fielded. Furthermore, performance shortfalls often led to increased costs, loss of political support, and a reduction in the number of aircraft to be procured.

The Air Force is in the midst of a transformational process to address the mismatch between the current force structure and a more responsive system architecture [434]. In the 2002 Air Force Posture Statement, former Air Force Chief of Staff General John P. Jumper said, "Our goal is to make warfighting effects and the capabilities we need to achieve them, the driving factor for everything we do. This enables (us to develop the capabilities needed) to answer a broad range of challenges posed by potential adversaries, while also developing the (assets needed) for the future" [231]. This shift has two primary elements. The first is that of *effects-based operations*[1], which relates to the actual employment of military systems [112]. The second is *capability-based acquisition* which directs the Air Force to procure capabilities instead of systems. The shift is explained by Col. Mike Holmes, Chief of the Air Force Strategy, Concepts and Doctrine Division at the Pentagon [174]:

[1] The concept of "effects-based targeting" was matured by Lt. Col. David A. Deptula during the 1991 Persian Gulf War. The concept has spawned a number of proponents and detractors over the past fifteen years.

> *"In the past, we sometimes started with a new (weapons) system that we could buy and then we tried to figure out what to do with it. What the chief of staff has asked us to do now is identify the effect we want to achieve on the battlefield and the capabilities required to achieve that effect. This requires us to determine what options are available to us – do we already have something in the inventory that can achieve this desired effect, or do we need to look for a new solution? This new solution may require purchasing a new weapons system, or it might be just finding a new way of doing business. It will require the Air Force to start with a problem and then determine what can be done to overcome that problem in order to accomplish the desired mission."*

The goal of capability-based acquisition is to field "militarily significant capabilities as soon as they become available" [463]. By constantly improving fielded systems through block upgrades, the military can keep up with threats that are constantly evolving. Dickerson notes that this process centers on the acquisition of a family-of-systems or system-of-systems that enables operations across one or more missions [134].

The fundamental shift toward capability-based acquisition and design is best described by a shift away from "things" to "ways to do things" as embodied in several key definitions of a capability:

> *"the **ability** to execute a specified course of action. A capability may or may not be accompanied by an intention"* [468].

> *"the combination of military equipment, personnel, logistics support, training, resources, etc. that provides Defence with the **ability** to achieve its operational aims"* [47].

> *"the **ability** to achieve an effect to a standard under specified conditions through multiple combinations of means and ways to perform a set of tasks"* [454].

The use of the term "capability" in the context of military planning first originated in the British Ministry of Defence and has also been incorporated in Australia and Canada [45]. The final oft-cited definition was developed by the Joint Chiefs of Staff in the formulation of the JCIDS. The word "ability," common to all three definitions, identifies the shift from stovepiped, service-centric, asset-based acquisition to a more flexible system that looks at the ways a mission can be accomplished. Because military objectives are often defined in terms of desired effects, the notion of a capability has a certain course of action or desired outcome in mind. The phrase "to a standard" implies that certain thresholds of desired of effectiveness are also tied to a capability. The standard defines how well a set of tasks must be performed. Different systems employed in different ways may provide the same capability to different standards. The "specified conditions" refer to the assumptions and the scenario in which the capability is employed. The employment of a system under conditions for which it was not designed may result in dramatically different consequences. Finally, the JCIDS definition of capability notes that multiple ways and means are employed in a combination of ways to achieve an effect. Complex interactions derived from the multiple potential combinations of ways and means confounds the analysis process and contributes both to design freedom and analysis challenges.

2.1.0.1 Capability-Based Planning is Not a New Concept

Also of note is a historical analysis capability-based planning concept. Early in the Cold War, the Department of Defense issued "General Operational Requirements" which were open-ended documents that specified the need for a new weapon system to fulfill a purpose. For example, General Operational Requirement No. 38 "called for an intercontinental bombardment weapon (a piloted bomber) that would replace the B-52 and stay in service during the decade beginning in 1965" [176]. This document was the genesis of the XB-70 *Valkryie*. General Operational Requirement No. 96 which "outlined the military need for three northern radar sites capable of detecting and tracking Soviet ICBMs" led to the Ballistic Missile Early Warning System (BMEWS) [396, 426]. Other General Operational Requirements resulted in systems such as the MIDAS early warning infrared satellite constellation, the

CORONA surveillance satellite, the U-2 surveillance aircraft, and the supersonic SR-71 [396]. These documents were primarily issued to address a military need when the specific physical implementation of a solution was not paramount. General Operational Requirements were also used when the technological advancements of the day were best understood by a handful of experts or when specific requirements were difficult to quantify due to the extreme technical risk of the programs undertaken. As technologies such as satellites and supersonic aircraft matured, the need to issue open-ended requirements statements were replaced by extremely specific requirements that required exact satisfaction for customer acceptance. Such arrangements were especially critical in multi-contractor competitions that sometimes involved litigation by the losing party. Today we see a shift back toward the more unconstrained approach, driven by the need to leverage new technologies and quickly adapt to changing threats and evolving capabilities.

2.1.0.2 Why Doesn't a Structured Process Already Exist?

Given the presence of policies that mandate a shift to capability-based acquisition, how is it possible that a structured method that supports JCIDS is not ubiquitous in the literature? While the Defense Acquisition University has published a number of documents on acquisition policy and the mechanics of JCIDS compliance, the mandate is still relatively new [124].

In March 2002, Secretary of Defense Donald Rumsfeld sent a memo to the Joint Chiefs of Staff directing a new way to evaluate requirements across a spectrum of alternatives [359]. The text of this memo, considered the birth of the JCIDS, is shown in Figure 8. The first revision of the JCIDS instruction (CJCSI 3170.01C) was published in June 2003. This was then superseded by CJCSI 3170.01D in March 2004 and CJCSI 3170.01E in May 2005 [370]. During this time, it is believed that many organizations have developed internal policies based on JCIDS but no major publications have been put forth to identify a consensus methodology.

March 18, 2002 7:17 AM

TO:	**Gen. Pace**
CC:	**Paul Wolfowitz** **Gen. Myers** **Steve Cambone**
FROM:	**Donald Rumsfeld**
SUBJECT:	**Requirements System**

As Chairman of the JROC, please think through what we all need to do, individually or collectively, to get the requirements system fixed.

It is pretty clear it is broken, and it is so powerful and inexorable that it invariably continues to require things that ought not to be required, and does not require things that need to be required.

Please screw your head into that, and let's have four or five of us meet and talk about it.

Thanks.

Figure 8: Genesis of JCIDS: March 2002 Memo from Secretary of Defense Donald Rumsfeld to Chairman of the Joint Chiefs of Staff General Peter Pace [220].

2.2 Quantitative Technology Evaluation

> *"For a successful technology, reality must take precedence over public relations, for Nature cannot be fooled."*
>
> -Richard P. Feynman

Technology evaluation is the assessment of the relative benefit of a proposed technology with respect to one or more capability-level metrics. Several government organizations including the National Aeronautics and Space Administration (NASA), the Defense Advanced Research Projects Agency (DARPA), the Office of Naval Research (ONR), the Army Research Laboratory (ARL), and the Air Force Research Laboratory (AFRL) are tasked with the identification and evaluation of advanced technologies for future application. For example, DARPA's mission statement is to "maintain the technological superiority of the U.S. military and prevent technological surprise from harming our national security by sponsoring revolutionary, high-payoff research that bridges the gap between fundamental discoveries and their military use" [125]. The AFRL is "a full-spectrum laboratory, responsible for planning and executing the Air Force's entire science and technology budget, basic research, applied research and advanced technology development" [423]. These research entities, by definition, have traditionally focused on the maturation of low readiness technologies which have the potential for high-payoff, asymmetric capabilities that change the fundamental dynamics of how systems operate.

It is difficult, in practice, to tie technologies with high uncertainty and revolutionary capabilities to direct application on existing platforms: a platform that uses a proposed technology may not even exist today. Furthermore, the ubiquitous nature of information across the globe means that new technologies are simultaneously available to all mankind[2]. This is a stark contrast to the historical pace of technology proliferation, for example, "the western world had not heard of gunpowder at a time when it was being used in China" [234].

[2]The spread of technology into society is referred to by Luce as *technology diffusion* [263].

These organizations are at an even greater disadvantage with the shift to a capability-focus. Under this paradigm, the functions a proposed system may perform may be analyzed at the early stages of design. While the addition of these degrees of freedom provides the ability to more thoroughly explore non-traditional means of providing a capability, it also confounds the technology evaluation by increasing the available design space without bound.

In contrast to a bottom-up exploratory forecasting approach where candidate technologies are proposed and their effectiveness is assessed against one or more measures of merit, a capability-based approach must examine the problem from a top-down view as shown in Figure 9.

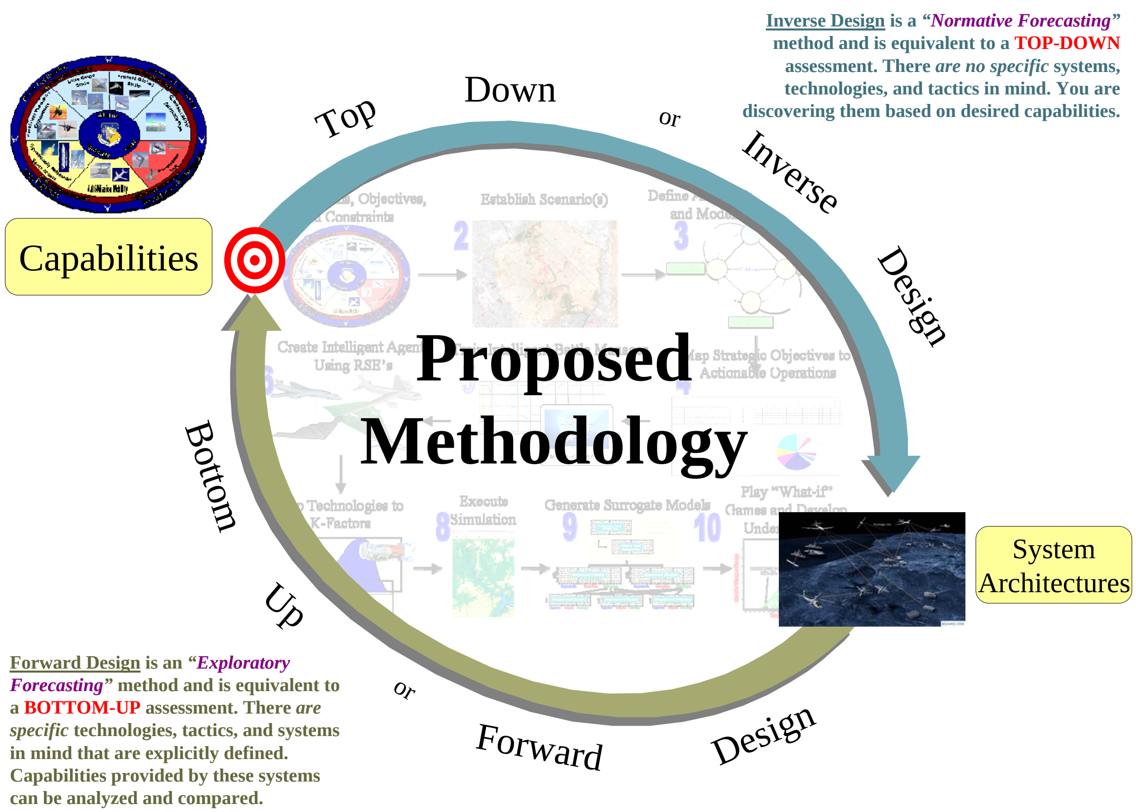


Figure 9: Relationship Between Forward Design and Inverse Design.

After defining one or more capabilities, the design space must be focused using systems engineering techniques to identify one or more system architectures to be examined. Next, a technique is needed to ***quantitatively*** explore the technology aspiration space with respect to capability-level MoEs.

A top-down approach is used to decompose capabilities into systems. A bottom-up approach is used to assess the capability gaps of one or more systems and close the loop between technology discovery and technology evaluation.

2.3 Introduction to Systems-of-Systems

While technology evaluation methods for systems have become more popular over the last several years, there has yet to be an emergence of a dominant technique for performing such assessments for *systems-of-systems*. The subsequent sections introduce a terminology for systems-of-systems and delineate between systems and architectures for the purpose of defining a nomenclature for use throughout this dissertation.

2.3.1 What are Systems?

> *"Life was simple before World War II. After that, we had systems."*
>
> - Admiral Grace Hopper (1906-1992)

A system, from the Greek *sunistanai* meaning "to combine," is "a combination of interacting elements organized to achieve one or more stated purposes" [217]. Systems can be generally categorized as either simple or complex. Simple systems, examples of which include oscillators and pendulums, can usually be described using closed-form analytical expressions and thus exhibit predictable behaviors. On the other hand, complex systems are often comprised of multiple, heterogeneous, interrelated elements and are difficult to study mathematically [260]. Complex systems are ubiquitous in engineering applications and demand a multidisciplinary approach to their study. Examples of complex systems include electrical distribution grids, a network of related computer software and data transmission devices, weather, and the human body.

A key aspect that distinguishes a *complex system* from a simple system or *component* is that a system generally has the property of emergent behavior: the combination of elements reveals a new function that the parts cannot provide in isolation. Bar-Yam notes that emergent behavior results when "the behaviors of many simple parts interact in such a way that the behavior of the whole is complex" [54].

In the context of systems, "complex" and "complicated" are not synonyms: while *complicated* tends to refer to large systems with many components, *complexity* is derived from the interwoven and interconnected nature of complex systems. Unlike simple systems that can be studied through Newtonian mechanics and decomposition, the interactions within a complex system are sometimes more important than the characteristics of the system components themselves. For this reason, a decomposition-based approach that ignores the strong interactions between system elements is inappropriate for the study of complex systems.

Although many subdisciplines have emerged to develop certain types of systems, the overarching scientific study of systems and their behavior is called *Systems Engineering*. Systems engineering is "a standardized, disciplined management process for development of system solutions that provides a constant approach to system development in an environment of change and uncertainty" [130]. A major focus of systems engineering is the development of total systems solutions, including supportability, operations and training, that satisfy customer requirements while balancing cost, schedule, performance, and risk. Systems engineering combines technical aspects of design with management techniques to ensure that these criteria are met. The systems engineering definition of system is particularly relevant to this research: "an integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective" [130].

The DoD 5000 series of directives on acquisition policy also identifies the need for systems engineering in acquisition, primarily to transform operational needs and requirements into a system solution that fulfills customer needs throughout the life cycle. It also ensures the compatibility and interoperability of the disparate elements in large-scale military systems and uses science and engineering to identify risk areas and mitigate them [452].

2.3.2 The Challenge of Heterogeneity

The primary difficulty with assessing military capabilities is not the hierarchical nature of the modeling requirements, but rather the complex interaction between the components of a system architecture. The Air Force alone has over 7,500 aircraft of 45 different types with many variants, derivatives, and blocks [423][3]. These systems must interact with each other, command and control entities, communications satellites, joint forces, and with coalition countries that may communicate using different languages, frequencies, standards, and units of measurement. Interactions between these elements can be temporary, evolutionary, or unpredictable, complicating the analysis of heterogeneous system architectures.

The primary difficulty in designing multidisciplinary systems with complex interactions and many components is the evaluation of different options that provide the same overall capability. Different architectures provide a different level of effectiveness and varying levels of cost, technology, and time to implementation. Currently, many elements of an architecture are optimized in isolation and seen as ideal by the members of their respective design organizations. There is no structured methodology for the comparison of dissimilar systems against the same top-level measures of effectiveness using constant assumptions.

2.3.3 Characteristics of Systems-of-Systems

In recent years, the term "system-of-systems" (SoS) has become increasingly popular terminology for a large-scale system that is comprised of a variety of heterogeneous, interoperable, collaborative systems[4] [27]. While the precise origin of this term is unclear, a 1964 paper by Berry on New York City refers to "cities as systems within systems of cities" [56, 250]. "The term 'systems-of-systems' is generally used to define a class of systems wherein a set of independent systems, each having unique behavior and performance, is organized to perform collaboratively and coherently to achieve a purpose" [110]. Definitions for system-of-systems abound in the literature:

- The Department of Defense defines a "system-of-systems" as "a set or arrangement of systems that are related or connected to provide a given capability" [370].
- The International Council on Systems Engineering (INCOSE) refers to the definition by Krygiel: "a system-of-systems is a set of different systems so connected or related

[3]There are also nine major commands, 35 field operating agencies, 352,000 active duty members and 423 active facilities around the world [16, 423].

[4]Since a system is comprised of parts, and a system-of-systems is comprised of systems, it is true that a system-of-systems is in fact a system.

as to produce results unachievable by the individual systems alone" [217, 250].

- The Air Force Scientific Advisory Board defines a "system-of-systems" as "a configuration of systems in which component systems can be added/removed during use; each provides useful services in its own right; and each is managed for those services. Yet, together they exhibit a synergistic, transcendent capability" [36].

Systems engineering fundamentals generally hold for larger-scale systems-of-systems except that interfaces are more difficult to define, coupling between systems is generally more complicated and less direct, and the problem of designing a system-of-systems is a very large scale effort that often requires extensive collaboration between geographically separated design entities [235].

Maier [267] identifies five principle characteristics in distinguishing complex systems from systems-of-systems that are widely cited in the literature[5]:

1. **Emergent Behavior**: The system performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire system-of-systems and cannot be localized to any component system. The principal purposes of the systems-of-systems are fulfilled by these behaviors.

2. **Evolutionary Development**: The system-of-systems does not appear fully formed. Its development and existence is evolutionary with functions and purposes added, removed, and modified with experience.

3. **Operational Independence of the Elements**: If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. The system-of-systems is composed of systems which are independent and useful in their own right.

4. **Managerial Independence of the Elements**: The component systems not only can operate independently, they do operate independently. The component systems

[5]Order changed

are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems.

5. **Geographic Distribution**: The geographic extent of the component systems is large. Large is a nebulous and relative concept as communication capabilities increase, but at a minimum it means that the components can readily exchange only information and not substantial quantities of mass or energy.

The first two characteristics are valid for *systems* as well. Specifically, all systems are designed to provide an emergent behavior. This emergent behavior is the useful outcome of component integration that can be used to provide capabilities.

Some *systems* also also developed incrementally or in blocks. An example of evolutionary development can be seen with derivative aircraft that have features added or removed over time. The next three characteristics are unique to large-scale systems-of-systems.

First, the elements of a system-of-systems have operational independence. If an aircraft (system) is decomposed into its subsystems, the avionics and engines don't *do* anything without the rest of the aircraft. On the other hand, a classic example of a system-of-systems is the national air transportation system which is comprised of aircraft, air traffic control facilities, runways, baggage handlers, ticketing agents, fuel trucks, and airports of varying sizes [207]. If an aircraft is removed from the air transportation system, both can still operate. The aircraft in turn can also operate independent of baggage handling and certain airports. This is not true of an aircraft system as it cannot operate if an engine is removed and vice versa.

This example also demonstrates managerial independence. Aircraft are controlled by their pilots, and guided by air traffic control. If radio contact is lost, the pilots can usually find an airport and land. If the digital engine control on an engine fails, it ceases to operate, regardless of whatever guidance may be provided by the pilot.

Finally, systems-of-systems are usually geographically distributed and their emergent behavior is derived by the exchange of information. In a national transportation system-of-systems, the primary means of coordinating interaction and deriving the emergent behaviors is by communicating with the various systems and providing instructions.

Maier's criteria are often used to definitely identify a system as a system-of-systems; however, ten years after his publication, disagreements bristle regarding the classification of systems as a systems-of-systems, families-of-systems, federations-of-systems, complex systems, complex adaptive systems, coalitions of systems, collaborative systems, interoperable systems, netcentric systems, supersystems, and others [342]. The INCOSE has recently offered another set of challenges that are unique to systems-of-systems [217]:

1. **System Elements Operate Independently**: Each system in a system of systems is likely to be operational in its own right.
2. **System Elements Have Different Life Cycles**: SoS involves more than one system element. Some of the system elements are possibly in their development life cycle while others are already deployed as operational. In extreme cases, older system elements in a SoS might be scheduled for disposal before newer system elements are deployed.
3. **The Initial Requirements are Likely to be Ambiguous**: The requirements for a system of systems can be very explicit for deployed system elements, but for system elements that are still in the design stage, the requirements are usually no more explicit than the system element requirements. Requirements for SoS mature as the system elements mature.
4. **Complexity is a Major Issue**: As system elements are added, the complexity of system interaction grows in a non-linear fashion. Furthermore, conflicting or missing interface standards can make it hard to define data exchanges across system element interfaces.
5. **Management Can Overshadow Engineering**. Since each system element has its own product/project office, the coordination of requirements, budget constraints, schedules, interfaces, and technology upgrades further complicate the development of SoS.
6. **Fuzzy Boundaries Cause Confusion**. Unless someone defines and controls the scope of a SoS and manages the boundaries of system elements, no one controls the definition of the external interfaces.

7. **SoS Engineering is Never Finished**: Even after all system elements of a SoS are deployed, product/project management must continue to account for changes in the various system element life cycles, such as new technologies that impact one or more system elements, and normal system replacement due to preplanned product improvement.

According to these guidelines, many military systems are systems-of-systems whose complexity is compounded by the fact that military procurement agencies seldom buy the entire system-of-systems at once, preferring a spiral approach that forces interoperability with legacy systems during the initial spirals.

While the delineation between a system-of-systems and a system can be difficult to identify, the classification between systems and subsystems can be equally difficult. To an airline, an aircraft is viewed as a system and an engine is viewed as a subsystem. To an engine manufacturer, the engine itself is a system comprised of subsystems such as turbines, compressor blades, and fuel nozzles. The scope of this challenge is summarized by Hatley: "every system below the level of the whole universe is a component of one or more larger systems. The larger systems are the context or environment in which the component system must work" [193].

While Lewe [258] identifies the level of certain systems hierarchically[6], an alternative definition is to understand a system as that which you are, a subsystem as that which you require to function, and a system-of-systems as that in which you belong. This "three-level sliding scale" is dependent on the view of the user, noting that most entities cannot see or understand much beyond their own field of view. An example of one such hierarchy for military system-of-systems analysis is shown in Figure 10. Aircraft and engines are placed between levels to underscore the message that the selection of labels is dependent on your point of view.

[6]Lewe notes the identification of system levels using Greek letters is credited to Robert Calloway.

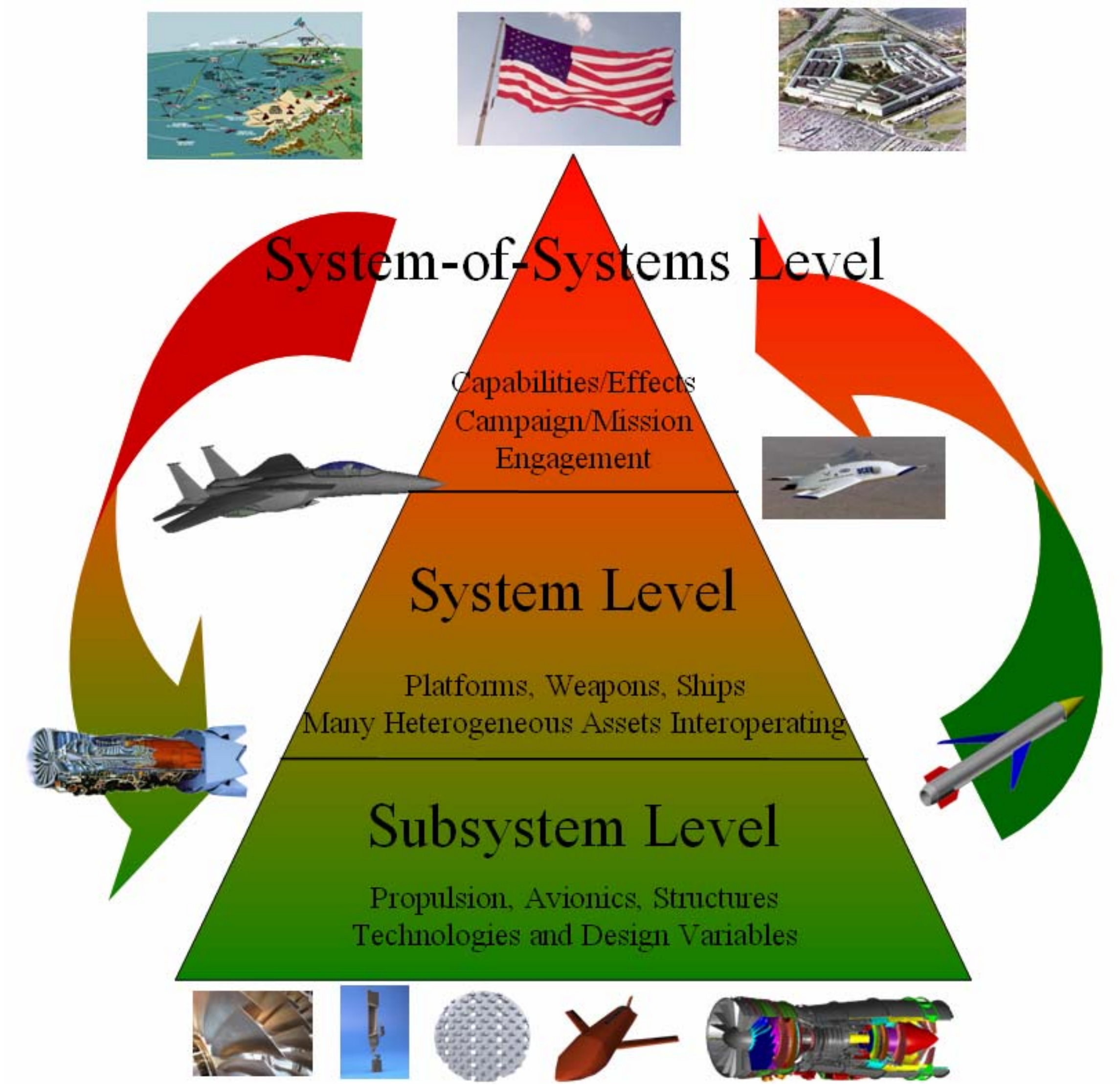


Figure 10: System-of-Systems, Systems, and Subsystems View of Military Architectures.

2.4 *Application: A Next-Generation Long Range Strike System Architecture*

Long Range Strike (LRS) capability, introduced in Section 1.3 as a suitable example application for a capability-based technology evaluation methodology, can be defined as "the ability to conduct prompt, accurate, conventional strikes anywhere on the globe on very short notice" [477]. The concept is also defined by the AFRL as the ability to "achieve a desired effect(s) rapidly and/or persistently, on any target, in any environment, anywhere, at any time." [444]. According to the 2006 Air Force Posture Statement, "responsive capabilities will combine speed, stealth, and payload to strike hardened, deeply buried, or mobile targets, deep in enemy territory, in adverse weather and with survivable persistence" [305].

Current Air Force Doctrine defines the LRS as an element of the Global Strike Task Force (GSTF) [230]. In this context, the primary objective for LRS assets is "kicking down the door" to open the way for the rest of the U.S. military. According to former Air Force Chief of Staff General John P. Jumper, the GSTF "is not necessarily your war-winning force. It creates the conditions for the war-winning force to get close enough to do their job" [331].

The primary motivation for future long range strike systems is that they address a critical shortcoming in the existing strike force. Currently designated "long range strike" systems (see Section 2.4.2) were primarily designed against Cold War threats. They are exceptionally effective against conventional military, economic, and infrastructure targets. As previously noted, rogue states and terrorist groups have a notable lack of targets for our existing systems. Elements such as weapons of mass destruction have an extremely high value to terrorist enemies: uncertainty about their location is a force multiplier and a large deterrent ability is required to avoid their use. This dichotomy between what our capabilities are and what they should be in the face of these threats is illustrated in Figure 11.

Currently, the U.S. military is very effective against traditional targets such as economics and infrastructure while lacking a robust capability against things which rogue states and terrorist groups hold in high regard. A long range strike system is designed primarily

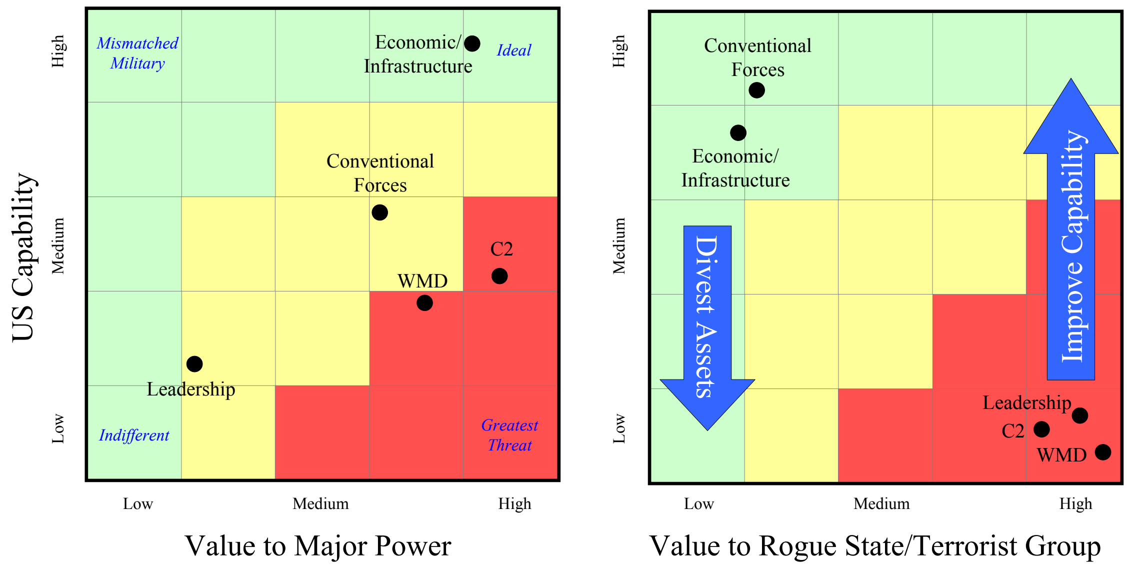


Figure 11: Comparison of Existing US Capability Against Enemy Value (Based on Reference [326]).

to provide capability against C2, leadership targets, and Weapons of Mass Destruction (WMD) facilities although it may also be effective in a traditional role. Subsequent sections summarize current and future LRS systems. These next-generation systems should be able to provide rapid, persistent, penetrating strike options in a denied-access environment in the presence of limited basing options.

According to the Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, desired capabilities for a future LRS system-of-systems include [479]:

- **Long-range**: Global from the continental U.S. (CONUS) or forward operating bases
- **Persistent**: 24/7 capability in anti-access environment
- **Responsive**: Respond globally within hours to minutes
- **Flexible, Precise Weapons Payload**: Mixed load, nuclear capable
- **Highly Survivable/Self-Defending**: Reduces support
  - Low observable, standoff weapons, speed, altitude
  - Manned, unmanned, or optionally manned
- **Global Situational Awareness**: Robust, fused sensor suites
- **Real-Time, Robust Beyond Line of Sight Connectivity**: Fully netted

- **Autonomous Operations**: Onboard sensors, offensive, defensive, non-traditional ISR
- **Flexibility/Adaptability**: Easily incorporate new capabilities, open architecture

The first five attributes are used to qualitatively examine existing and proposed LRS aircraft with respect to capability gaps (see Figure 17). The next four attributes are desirable features of an LRS architecture: no present-day systems possess these attributes. A process for quantitatively assessing potential LRS architectures with respect to these metrics is defined in a subsequent section.

2.4.1 The History of Long Range Strike Systems

Range has always been a defining factor in military operations, and the technology to extend the range of weapons has often led to revolutionary capabilities on the side of the aggressor.

Without technology, projectile attacks are essentially limited to the distance over which a human can hurl a projectile (about 30 meters). Siege weapons such as the catapult emerged during Greek times (around 400 B.C.) and were used by Alexander the Great for battlefield cover in addition to attacking hardened fortifications. Other variations including the onager, trebuchet, and ballista which were capable of hurling stone projectiles to distances up to around 300 meters [271]. Artillery emerged as the dominant technology in the 14th century with the proliferation of gunpowder. Cannon of the 16th and 17th century extended the strike range of military forces to several thousand meters [269]. Modern *Paladin* howitzer self-propelled field artillery pieces are capable of launching a standard shell about 18 km and a rocket-assisted shell over 30 km but are limited in their mobility and speed [16]. The largest gun ever produced was the German "Paris Gun" which was able to launch a 94 kg shell over a distance of up to 130 km, but required an extensive support system and rail mountings for transportation.

The advent of the battleship combined the lethality of long-range ground attack with the mobility of ocean-going vessels. The U.S. *Iowa*-class battleships have nine 16-inch guns that can fire a 1,225 kg projectile nearly 39 km [16]. Since many of the world's population centers and military facilities are located near coastlines, battleships provided a cost-effective means of holding these targets at risk until their decommissioning in the early 1990's. Despite their utility in the fire support role, the power projection capabilities of the battleship were largely outstripped by the flexibility and mobility of aircraft operating from sea-based carriers [97]. Naval aircraft extended the range of strike systems to several hundred kilometers. This capability is complimented by ship and submarine-based cruise missiles which essentially hold at risk any target within about 1,600 km from shore [464].

While ground-based aircraft are limited to fixed base locations, the advent of aerial refueling in the late 1940's extended the range of aircraft nearly indefinitely, enabling global reach for the first time. Heavy bombers from the 1950's and 1960's have ranges that exceed 8,000 km [422]. If refueled over international airspace, long range bombers can attack nearly every target on the surface of the globe.

The advent of ballistic missiles in the 1940's opened the ultimate high ground for long range strike systems. The V-2 missile, produced by Germany during World War II had a range of approximately 300 km and could deliver a 1000 kg warhead with a 50% probability of being within 17 km of the intended target [16]. In contrast, the Minuteman III ballistic missile which entered service less than 20 years after the introduction of the V-2 can deliver a similar payload to a distance of over 10,000 km with a hundred times greater accuracy [423].

The exponential development of weapon system capability through technology infusion has increased the range of combat from the limits of hand-thrown projectiles to weapons that can deliver large payloads with global reach. Technological shortfalls still exist in response time, lethality, accuracy, and the reduction of collateral damage.

2.4.2 Current "Long Range Strike" Systems

In 2001, The U.S. Air Force released a document called "U.S. Air Force Long-Range Strike Aircraft White Paper" [24]. Curiously, this is an update of a 1999 document entitled "U.S. Air Force White Paper on Long Range Bombers" [21]. Recently, the Air Force equates long range strike capability with the three active heavy bomber systems that provide "long range strike" services to the military: the venerable workhorse B-52 *Stratofortress*, the rugged supersonic B-1B *Lancer*, and the stealthy flying-wing B-2A *Spirit* (Figure 12). The document was altered primarily to support the transformational shift of the LRS mission to the Global Strike Task Force (GSTF), an expeditionary force that "kicks down the door" to clear the way for other U.S. and allied assets. Unfortunately, "U.S. forces arriving in a theater in the opening days of a major conflict will be badly outnumbered" [385]. To be successful, the GSTF must leverage the best technology, tactics, and training to gain a qualitative advantage over a numerically superior force.

Figure 12: Air Force Long Range Bomber Force (Adapted from [24]).

2.4.2.1 B-52 Stratofortress

The B-52 bomber first entered service as a long range nuclear bomber in February 1955. Of the 744 B-52's built, 102 B-52H models were delivered to the Strategic Air Command[7] between May 1961 and October 1962 [148]. The unrefueled range of the aircraft exceeds 14,172 km (7,652 nm) with a maximum speed of over 1000 kilometers per hour (Mach 0.86) and a ceiling of 15,240 m (50,000 ft) [422]. Its payload capacity is 31,752 kg (70,000 lbs) and it has the ability to hold 45 weapons (27 internal and 18 on externally mounted pylons). The H model can carry over 32 types of conventional or nuclear munitions including the JDAM, JASSM, sea mines, and air-launched cruise missiles [7]. Up to eight AGM-88 cruise missiles can be carried internally on a rotary launcher and another six can be mounted on wing pylons [202]. Though the B-52H airframe is over 40 years old, recent upgrades

[7] now Air Combat Command

allow the aircraft to utilize advanced electronic offensive and defense systems, function at extremely low-altitudes with terrain-following radar, and operate over longer ranges due to upgraded Pratt & Whitney TF-33 turbofan engines [401]. Following the retirement of the B-52G model in 1993, there are 94 aircraft in the inventory, of which 44 are combat coded [408]. Despite certain opposition from Congress, the Air Force has recently indicated a desire to decrease the number of B-52's to 56 airframes [205].

2.4.2.2 B-1B Lancer

Design studies for the B-1 bomber began with the Advanced Manned Strategic Aircraft program in 1965. Following the cancellation of the North American XB-70 *Valkryrie* program in 1969, the North American Rockwell (now Boeing) B-1 was conceived as a supersonic strategic penetrator with a top speed in excess of Mach 2. Its primary weapon was the nuclear AGM-69A SRAM. The CONOPS for the B-1 relied on a mix of subsonic/supersonic flight as the Soviet surface-to-air-missile (SAM) systems of its era were primarily calibrated for medium to high altitudes. "Studies of the period showed that the best chance of successful penetration of a heavily defended area lay in high subsonic speed at low altitude combined with Mach 2 performance at high altitude to reduce transit time through lightly defended areas" [380]. This assessment led to the swing-wing design of many aircraft of this period. After a very successful flight test program, President Carter cancelled the B-1 program in 1977 in favor of the new "wonder weapon," the cruise missile. Due to anticipated delays and technical risk associated with the Advanced Technology Bomber (later the B-2), President Reagan reinstated the B-1 program on October 2, 1981 announcing the acquisition of 100 aircraft to serve as interim solutions to America's bomber shortage. There were several notable differences between the B-1A and B-1B programs. First, the Mach 2.2 cruise speed was reduced to Mach 1.25. This allowed less complicated inlets which had the added benefit of decreasing radar cross section. The maximum takeoff weight of the aircraft was increased from 179,170 to 216,364 kg (395,000 to 477,000 lbs) and Radar Absorptive Materials (RAM) were applied to the aircraft to counter the main threat for low level penetrators: fighter aircraft with down-looking radar [380]. Finally, the B-1A's two-plane radar system was replaced with the single electronically-steered Westinghouse APG-164. This system featured intermittent pulses instead of continuous operation and further decreased Radar Cross Section (RCS). It is said that the B-1B had an order of magnitude reduction in signature over its predecessor.

After the end of the Cold War the B-1B was transitioned to carry conventional munitions. The B-1B, through avionics upgrades, is the only bomber that can carry three types of weapons simultaneously in its three weapons bays. The $200 million dollar bomber has the largest payload capacity of any US bomber, 34,019 kg (75,000 lb), and its payload can be increased to 56,700 kg (125,000 lb) with external carriage although this is disallowed by the START I treaty [16, 380]. Over ten types of weapons can be employed including up to eighty-four 500 lb GBU-30 JDAMs or twenty-four AGM-158 JASSMs. This heavy bomber has an unrefueled range of 12,000 km (6,479 nm) although the range is reduced to 5,543 km (2,993 nm) with a standard weapons load [5]. Though some critics believe the B-1 is redundant, its supersonic capability makes it the only bomber with reasonable response time for time critical targets (TCTs). The B-1 has been used to great effect in Operation *Iraqi Freedom* using a tactic where it loiters subsonically outside enemy airspace and dashes into the battlespace supersonically for close air support when called by ground forces [478]. Though there is some continuing debate about the required number of B-1 bombers, there are currently 67 aircraft in the active force, all of which are combat coded [423].

2.4.2.3 B-2A Spirit

The most advanced heavy bomber in the world is the B-2A *Spirit* Stealth Bomber. Beginning as a “black program” called the High Altitude Penetrating Bomber (later Advanced Technology Bomber or ATB) in the 1980’s, the unique flying wing design is based on early Northrop designs such as the propeller-driven XB-35 and the jet-powered YB-49. Although Northrop lost the F-117 contract to Lockheed in the late 1970’s, experienced gained with stealthy, curved aerodynamic shapes in the 1978 *Tacit Blue* program contributed to design experience for the B-2 [380]. Due to the extreme secrecy of the ATB program, there was little opportunity for public criticism of the massive development cost of the bomber. According to the Government Accountability Office (GAO), by 1997 the total program cost of the B-2 bomber approached $45 billion dollars [354]. Originally conceived as a replacement for the aging B-52 in the nuclear attack role, the B-2 is currently the only US aircraft with a long-range capability to penetrate defended airspace [186]. No country has either an equivalent or an effective defense against the B-2 and it is considered the ideal first-strike weapon against fixed targets in all weather conditions anywhere in the world. Its ability to deploy on strike missions from Missouri with little logistics support make it the only US system, other than submarine launched cruise missiles, that can effectively use the element of surprise. With the exception of its low cruise speed (Mach 0.85), it is arguably the closest aircraft system to the capability statement for Long Range Strike.

In the early 1980s, threat projection experts decided that low-altitude penetration was the key to avoiding Soviet radar systems (see section 2.4.2.2) due to the masking properties of the Earth's curvature. For example, a radar that can detect an aircraft flying at 8,000 meters altitude from 370 km has only a 23 km range when the same aircraft operates at an altitude of 30 meters [386]. Although the B-2 was less visible at high altitudes than its predecessors, rapid advancement in Soviet radar systems revealed that it was only a matter of time before radar systems could track stealth aircraft. As a result, during the design process for the B-2, threat projectionists dictated that the bomber should use low-altitude maneuvering. The large wing area and low wing loading of the B-2 did not make it ideal for this mission because structural flexing would alter the RCS and destroy the stealth characteristics [380]. Extensive redesign was undertaken, further driving up the cost. In practice today, the B-2 operates primarily at high altitudes although it was designed to fly at very low altitudes. This example underscores the importance of Frits' hypothesis that tactics should be developed concurrently with system design [159].

The B-2 has a maximum takeoff weight of around 154,211 kg (340,000 lbs) and can carry 18,143 kg (40,000 lbs) of munitions. In Operation *Allied Force*, the B-2 was the first aircraft to demonstrate the GPS-guided JDAM in combat [467]. Although the B-2 is capable of in-flight refueling, its unrefueled range is generally assumed to be approximately 12,000 km

(6,500 nm) although some estimates are as high as 14,800 km (8,000 nm). Supporters of the $2 billion dollar aircraft cite reduced *life-cycle costs*: the high survivability afforded by the stealth design eliminates the need for an armada of support aircraft to provide standoff jamming and destruction of enemy air defenses. There are currently 21 B-2 bombers, 16 of which are combat coded [408].

2.4.2.4 F-22A Raptor

Although it is not included in the 2001 White Paper on long range strike systems, the designation of the F-22 *Raptor* was changed to F/A-22 at the 2002 Air Force Association National Convention. Former Air Force Chief of Staff General John P. Jumper said, "the change is meant to more accurately reflect the aircraft's multimission roles and capabilities in contemporary strategic environments" [170]. In December 2005, the F/A-22 was redesignated F-22A when it entered service to reflect the importance of its original mission. Although it was originally intended as an air dominance fighter and is currently employed in this manner, this extremely advanced, maneuverable, and stealthy aircraft can also provide limited long range strike capability. Its internal weapons bay was primarily designed for the air-to-air combat role, but the F-22A can also be outfitted with two 1,000 lb GBU-32 JDAMs along with two AIM-120C AMRAAMs in the main bay and two AIM-9 Sidewinder missiles in the side bay [170]. The top speed of the F-22A is classified (estimated to be at least Mach 1.8); however, its high thrust-to-weight ratio Pratt & Whitney F119-PW-100 engines are designed to allow the F-22A to cruise at at least Mach 1.5 without using afterburners[8] (Jumper flew the *Raptor* to Mach 1.7 without afterburners on January 12, 2005 [346]).

Although it is widely assumed that the F-22A can operate at Mach 1.5 for its entire mission, supercruise is only used for the last 90-180 km (50-100 nm) of the mission and greatly impacts range, as shown in Figure 13. This is still a large advantage over traditional afterburning aircraft that can typically only travel 18.5-37 km (10-20 nm) before exhausting their reserve fuel supplies. The maximum internal payload of the F-22A is approximately

[8] Termed "supercruise."

1,776 kg (3,915 lb), and it can carry approximately 8,618 kg (19,000 lb) of external payload although this would likely not be used for stealth operations [2].

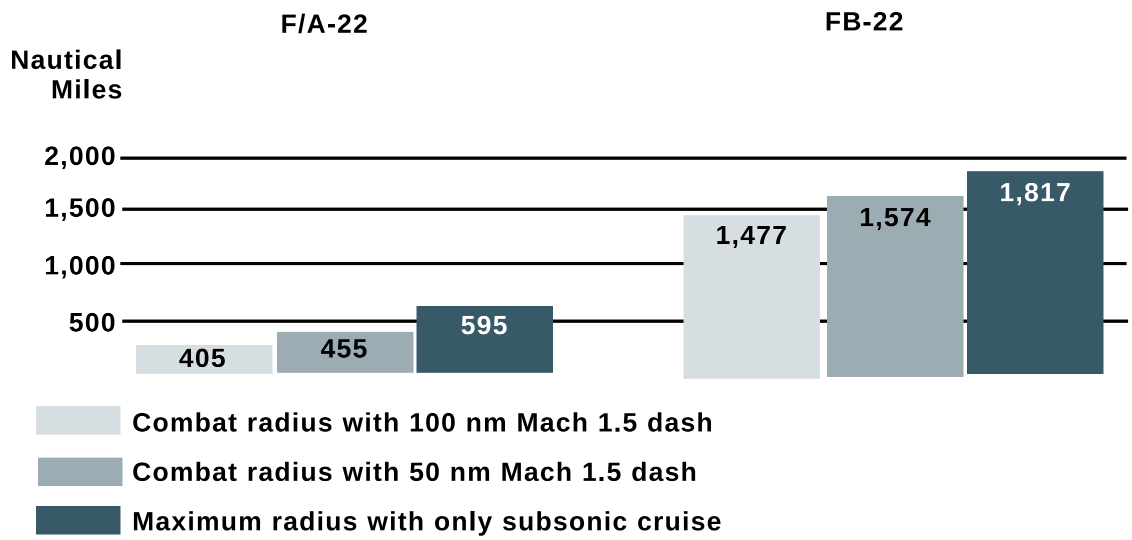


Figure 13: Difference in Mission Radius for F-22 Variants [410].

In April 2003, the Air Force issued a request for information on long range strike systems that could notionally be fielded by 2025. One of the proposed solutions is a variant of the F-22A called the F/B-22. This aircraft would be designed to carry at least 30 GBU-39 Small Diameter Bombs (the F-22A would carry eight), penetrate enemy airspace and persist with its enhanced survivability [409]. The concept has also been referred to as a "regional bomber" [201]. A comparison of the F-22A and the F/B-22 (see Figure 14) and their respective mission ranges for differing amounts of supercruise is shown in Figure 13.

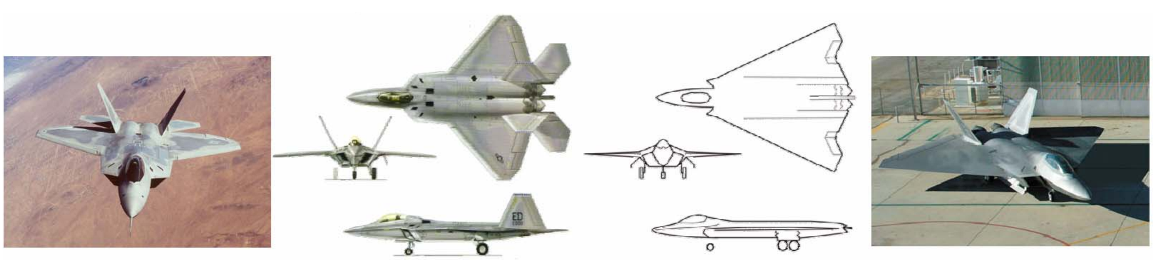


Figure 14: Comparison of the F-22A (Left) and F/B-22 (Right), Adapted From References [2, 410].

2.4.2.5 BGM-109 Tomahawk Cruise Missile

Military operations inherently involve cooperation between joint forces and international coalitions. This further complicates the interoperability issue and increases the difficulties faced by commanders. Nevertheless, although the Air Force is the primary branch of the US military with LRS capability, other branches can also provide support in this role. The Navy Tomahawk Land Attack Missile (TLAM) is a subsonic cruise missile deployed from U.S. Navy vessels and U.K. Royal Navy submarines. The current generation Block IV TLAM-E, or "Tactical Tomahawk" is manufacured by Raytheon Missile Systems. The $569,000 (FY99 $) missile has a range of 1670 km (900 nm) with a speed of about 885 kph (550 mph). The TLAM-E features the ability to loiter over a target area and assess battle damage or designate new targets using an on-board camera. This missile is the first Navy cruise missile that can be reprogrammed in flight to strike any of 15 pre-programmed alternate targets or redirect to any GPS coordinates [464]. The Navy plans to convert 1,253 anti-ship variant missiles to the TLAM-E configuration and procure over 3,000 more [149]. Navy vessels that can use the TLAM include the DDG-51 *Arleigh Burke*-class destroyers, DD-963 *Spruance*-class destroyers, CG-47 *Ticonderoga*-class cruiser surface ships and the SSN-688 *Los Angeles-class*, SSGN-726 *Ohio*-class submarines [14]. British Royal Navy SSN *Swiftsure*-class subs can also employ the weapon [99].

A cutaway of the Block IV Tomahawk cruise missile with the major Block III and Block IV upgrades listed is shown in Figure 15 and a summary of the estimated range of the above long range strike concepts and a comparison with other fighters and bombers is shown in Figure 16.

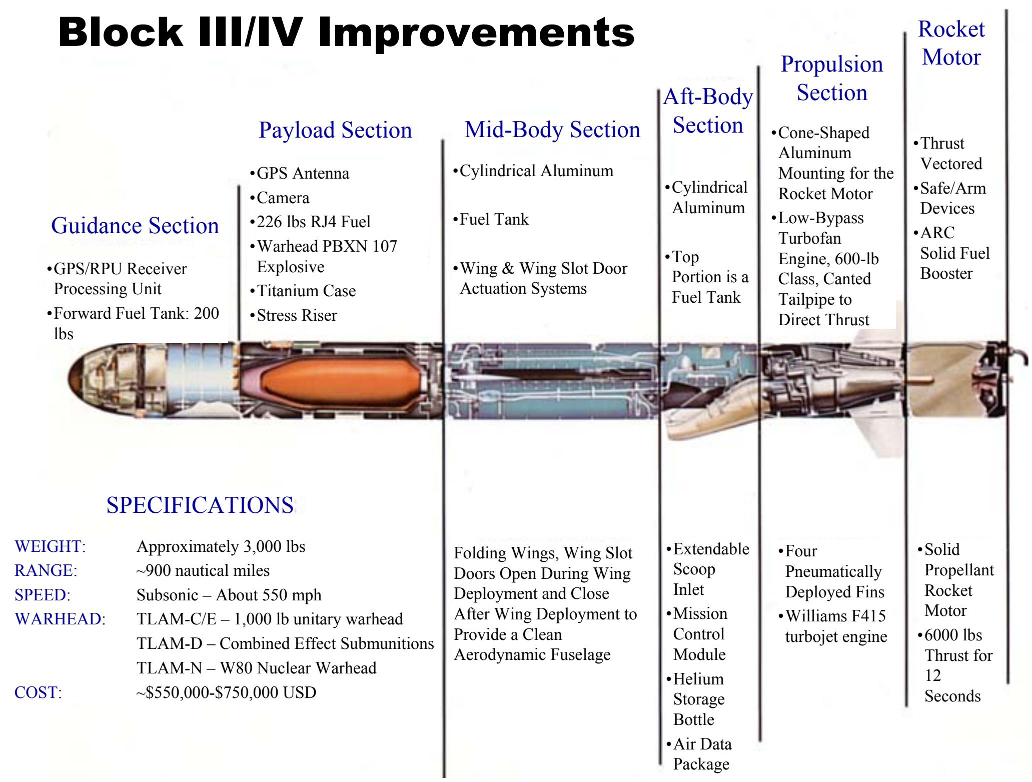


Figure 15: Tactical Tomahawk TLAM-E (Adapted From References [246, 464]).

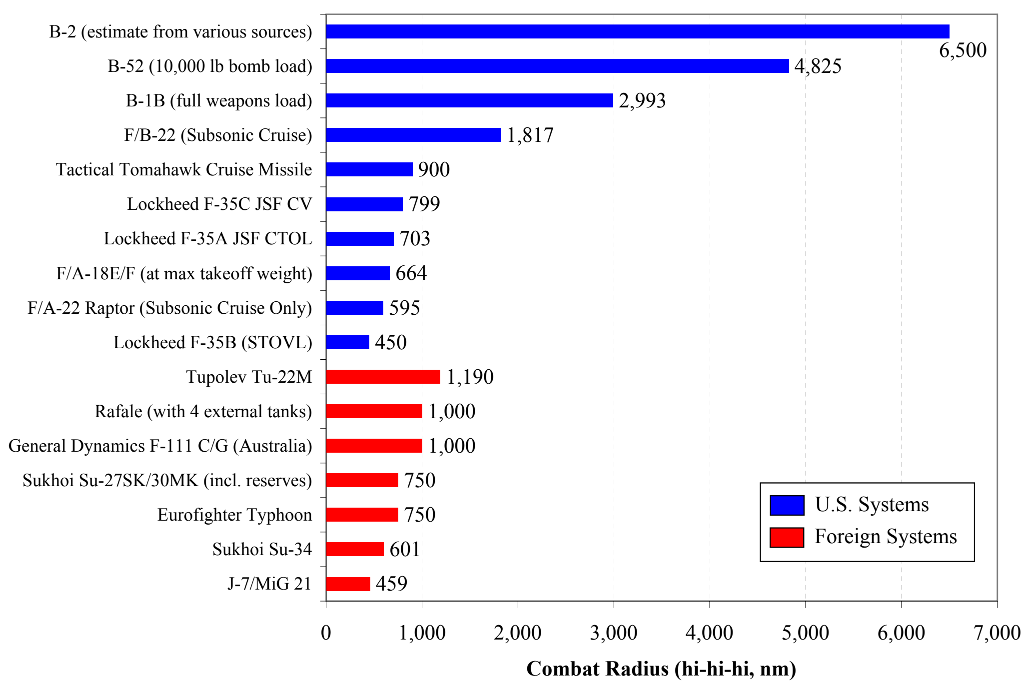


Figure 16: Combat Radius Summary for Foreign and Domestic Strike Systems, Compiled From References [2, 8, 10, 15, 246, 247, 410].

A qualitative assessment of the LRS attributes of the previously mentioned current LRS systems is shown in Figure 17. The figure qualitatively shows why the DoD has highlighted a capability gap in Long Range Strike. While several of the concepts have a global range from CONUS, few have the ability to persist in denied airspace (although the B-52H and B-1B have standoff weapon capabilities). The capability shortfall is most seriously manifested in the lack of responsiveness of current systems. While long-range bombers can deploy from CONUS, their subsonic speed constrains the response time to nearly a day and their support requirements severely restrict sortie rates. In addition to identifying compliance with the first five LRS attributes highlighted in Section 2.4, a qualitative assessment of life-cycle cost is also given in Figure 17. The definitions of the evaluation criteria used are given on Page 39.

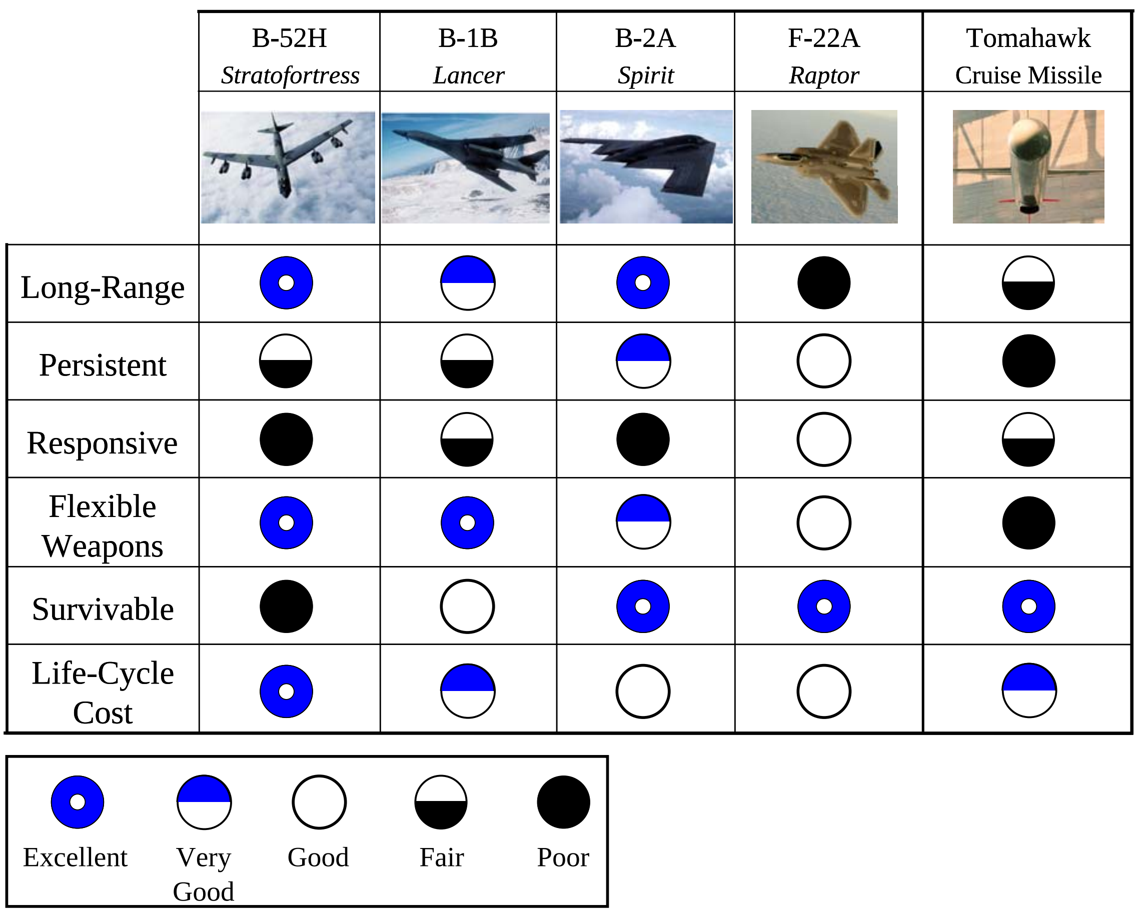


Figure 17: Current Long Range Strike Systems (Supported by Data from Reference [68]).

2.4.3 Future Long Range Strike Systems

The Long Range Strike White Paper identified the years 2035-2037 as the appropriate date for the IOC of a next-generation bomber [21]. By this date, attrition is likely to have reduced the fleet below the minimum 170 aircraft desired [169]. This date is also based on conservative peacetime estimates of bomber utilization, and the impact of continuous usage in Afghanistan and Iraq is unclear. Air Force leadership is now in the process of examining long range strike systems that could come online between 2015 and 2025 [201]. The 2006 Quadrennial Defense Review report and a recent article in *MarketWatch* identified a push to utilize existing technology to field a solution by 2018 [456, 95].

Over the past several years, a number of studies have examined alternatives to supplement Long Range Strike capabilities, in fact, as Thompson notes, "on average, one study of long-range strike requirements has appeared per fiscal quarter since the Cold War ended" [403]. The Future Strike Aircraft (FSA) program, a $1M study directed by the Aeronautical Systems Center at Wright Patterson AFB, Ohio identified several high-speed platform concepts capable of global strike missions. Deployment from CONUS was a requirement and the primary trade space identified for this study was the degree of speed or stealth required[9]. Vehicles with top speeds from Mach 2.7 to 14 were examined [169]. The Long Range Strike Aircraft (LRSA-X) study expanded upon FSA to examine LRS in a system-of-systems context. Supersonic through hypersonic solutions were examined for their ability to penetrate current and next-generation IADS to defeat time sensitive and hardened targets [171]. Also, the Long Range Strike Platform (LRSP) study aimed to identify technology investment areas in platform concepts, weapons systems, and C4ISR that enabled concept refinement for a future LRS system. In 2004, the Air Force planned to use a portion of a $45M Congressional plus-up to establish a program office for the analysis of future LRS concepts [172]. Several of the concepts identified from these studies are shown in Figure 18.

A subsonic penetrator is a vehicle similar to the B-2A *Spirit*. Its primary attribute is stealth. Two other subsonic platforms were identified. A "missileer," also known as an

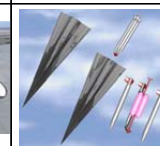
[9] The simultaneous application of both is anticipated to be very costly.

"arsenal plane" is a long-range, unstealthy vehicle, usually a cargo plane, that is designed to fire standoff weapons outside defended airspace. A variation on this concept is a wing-in-ground effect vehicle. These vehicles generate a high lift-to-drag ratio by flying very close to the ground or water to reduce the impact of tip vortices. Such vehicles, called *ekranoplanes*, were flight tested in the former Soviet Union. In 2003, Boeing proposed a wing-in-ground effect vehicle called the Pelican with a 152 m (500 ft) wingspan and a 1.3 million kilogram (2.8 million pound) payload [2]. As an LRS concept, these vehicles would fly subsonically under radar and launch cruise missiles or even other strike platforms. Supersonic bombers with speeds from Mach 2-4 were also examined. A major technical challenge in sustained supersonic flight is the development of materials: existing stealth materials are not well suited for high speed flight regimes and new coatings to mask infrared signatures in the presence of high heating rates would be required [478].

Reusable hypersonic cruise vehicles were also examined. Air Force General David Deptula noted that hypersonics offers "revolutionary responsiveness, reach, and range" [200]. A concept called the Common Aero Vehicle (CAV), a maneuvering reentry vehicle designed to attack deeply buried hardened targets by deploying submunitions as it reenters the atmosphere, is also being considered. The CAV is one of the elements of the DARPA/Air Force Application and Launch from the Continental United States (FALCON) technology demonstration program. A CAV could be launched using refurbished strategic missiles or deployed directly from low Earth orbit. Other orbital weapons and reusable spaceplanes that deploy precision weapons from orbit are also being considered [175]. Recently, the CAV has been redesignated to the "Hypersonic Technology Vehicle" (HTV) to shift the focus away from direct weaponization of the technology [131]. It is important to note that this list is by no means all-inclusive: the Air Force has performed at least 24 studies on LRS since 1999 and additional concepts may be under development in a proprietary or classified environment [132].

A qualitative comparison of proposed next-generation systems is shown in Figure 18[10].

[10] A 2006 Congressional Budget Office study generally supports the results of Figure 18 and proposes a first-order cost estimate for these systems [37].

	Near Term		Mid-Term			Long Term		
	Subsonic Penetrator	Missileer/ Arsenal Plane	Wing-in-Ground Effect	Low Supersonic (M~2)	High Supersonic (M~4)	Hypersonic	Suborbital	Orbital
								
Long-Range	Excellent	Excellent	Very Good	Good	Good	Very Good	Very Good	Excellent
Persistent	Very Good	Good	Fair	Fair	Fair	Poor	Poor	Very Good
Responsive	Poor	Poor	Poor	Fair	Good	Very Good	Excellent	Very Good
Flexible Weapons	Very Good	Very Good	Good	Good	Good	Fair	Fair	Poor
Survivable	Excellent	Poor	Poor	Very Good	Very Good	Very Good	Excellent	Very Good
Life-Cycle Cost	Fair	Excellent	Very Good	Good	Good	Poor	Fair	Poor
Low Technical Risk	Excellent	Excellent	Good	Very Good	Good	Poor	Fair	Poor

Excellent Very Good Good Fair Poor

Figure 18: Comparison of Air Force Generic LRS Concepts (Concepts from Reference [446], Rankings by Author).

While Watts and the Congressional Budget Office identify these concepts as potential options for LRS systems, the AFRL classifies the potential solutions in terms of near-term, mid-term, and long-term potential [101]. This research focuses on technology infusion to near-term and mid-term systems due to the technological uncertainty of long-term concepts and the dearth of high-fidelity models to support quantitative analysis of hypersonic vehicles and their respective propulsion systems, ballistic missile guidance systems and trajectories, and orbital constellations.

2.4.4 Summary of LRS Capability

Long Range Strike capability is gaining increasing interest within the acquisition community:

- The 2005 Air Force Handbook recognizes that the requirements for a future LRS system have yet to be defined and that LRS is pre-decisional and is within the framework of the JCIDS process [132].
- Current leadership suggests an interim approach phased for IOC in the 2015-2020 timeframe that can provide 24/7 stealth, deliver rapid and persistent effects against moving targets, hardened targets, and deeply buried targets, day or night and in all weather conditions [95].
- The 2006 Quadrennial Defense Review notes that the Department of Defense "begin development of the next generation long-range strike systems, accelerating projected initial operational capability by almost two decades." Goals for this activity include a 50% increase in capability, a 500% increase in penetrating capability[11]. DoD estimates that about 45% of the future LRS systems will be unmanned. [456]
- Recently, "the Air Force Studies Board of the National Research Council (NRC) was asked by the USAF to investigate combinations of speed and stealth that would provide U.S. aircraft with high levels of survivability against potential enemy air defense systems in the 2018 time frame. The missions considered were to include but not be limited to long-range strike" [311].

[11] The units of measure and standards of comparison for these goals are not given.

Acquisition authorities are currently conducting an analysis of alternatives to identify requirements for a Long Range Strike system [403], the AFRL has recently reopened a wind tunnel to support LRS aerodynamic studies [445], and the 2007 USAF Budget Request projects a dramatic increase in funding for Next-Generation Long Range Strike projects over the next five years (see Figure 19). Regardless of the concept chosen, advanced technologies will be infused into the system and its supporting architecture to prevent technology obsolescence as new threats to national security emerge. The urgent need to refine America's bomber fleet coupled with the myriad of technologies required across the system architecture to enable revolutionary LRS capabilities drives its selection as the proof-of-concept application for the proposed methodology.

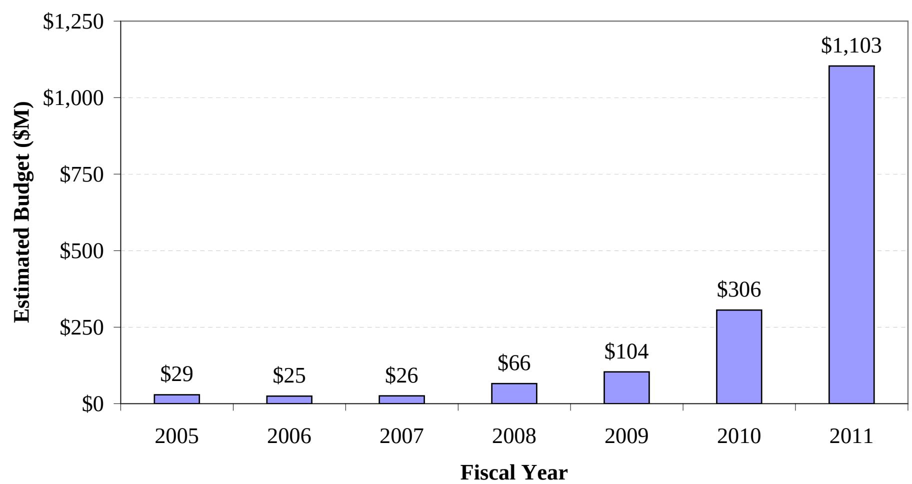


Figure 19: United States Air Force 2007 LRS Budget Request (Compiled from Reference [437]).

2.4.5 How Large of an Architecture is Needed to Demonstrate the Methodology?

History indicates that there are many ways to project strike power over long distances and recent LRS studies summarized in Figure 18 identify possible multi-domain solutions that may offer revolutionary capabilities; however, aircraft-based LRS solutions remain an area of interest for military planners. While aircraft may be one of the most vulnerable long range attack systems, they also offer benefits in terms of operational flexibility, response time, and payload capacity. For the proof-of-concept study, the LRS architecture under consideration is limited to solutions in the air domain; however, future work may extend the proposed methodology to demonstrate the approach with additional degrees of exploration.

In addition to the air vehicle which serves as the centerpiece of an LRS architecture, other enabling "architecture technologies" including communications, ISR, jamming support, C2, theater missile defense, and others contribute to the success of the LRS mission. One key constraint in the definition of an example problem is establishing an appropriate "control volume" to demonstrate the proposed methodology without inducing "analysis paralysis." To draw a control volume, it is necessary to classify the the level of heterogeneity in the LRS architecture to be studied.

While the proposed methodology addresses the need for technology evaluation for systems-of-systems, how much is enough? A system comprised of identical elements are of little interest as the complex interactions between systems define the major complexity issue that defines the need for a structured methodology. On the other hand, the design of communication systems, network architectures, new smart munitions, command facilities and the like is outside the scope of effort defined for this research. *Architecture technologies* related to the employment of the aircraft and the interfaces between these technologies and the LRS systems to be examined are the primary focus of this research effort. The specific technologies that will be examined in the architecture context are defined in Section 5.7. The acceptable degree of heterogeneity is in the ***middle*** of the spectrum between an entire system-of-systems and a singular system comprised of physically similar elements.

2.5 Recapitulation of Problem

The first step of the scientific method is problem definition. Through a review of non-technical literature available on capability planning, systems engineering, and technology assessment several key observations that focus the definition of the problem are made:

1. Technology evaluation is a resource allocation problem. Identification of high-payoff technologies is only critical in a *resource-limited* environment. Since fiscal resources are always limited, technology evaluation must balance performance and cost to maximize the overall benefit.
2. The JCIDS process is a policy mandated by DoD to address the need for a military architecture that is robust against constantly changing threats.
3. There is a general lack of structured processes and methods that are consistent with the JCIDS process. Efforts to comply with the policy are largely ad-hoc and vary from entity to entity.
4. A capability-based approach looks at the ways and means of performing an action and is not tied to a single physical solution.
5. Military systems-of-systems are dominated by heterogeneous, interoperating assets with different life-cycles that must be integrated to provide capabilities.
6. Systems-of-systems engineering is an emerging field rife with jargon and terminology that must be crisply defined as the field matures.
7. Existing system-of-systems engineering approaches and methods for technology evaluation do not rely heavily on quantitative analysis. This is primarily due to the confounding effects of the complex interactions across the hierarchical system of systems.
8. While a capability-based approach can be used to decompose strategic challenges into a variety of capabilities, Long Range Strike capability is a pressing issue facing the military acquisition community and has the necessary elements and interactions to demonstrate a methodology for capability-based technology evaluation
9. To limit the scope required analysis, potential solutions are confined to the air domain.

10. To further constrain the scope of the modeling and simulation activity to a representative experiment that does not confound the methodology proof with unnecessary overcomplication, a medium level of heterogeneity is proposed.

While the above statements focus the demonstration activity on a manageable problem that can be addressed using available resources, the original problem domain is actually much larger. An appreciation for the complexity of system studies can be observed using a matrix of alternatives or morphological matrix[12] as shown in Figure 20.

Observations and Assertions	1	Level of Financial Resources	Limited	Unlimited		
	2	Acquisition Focus	Requirements Generation System	DoD 5000	Joint Capabilities Integration and Development Sys	Other
	3	Level of Process Structure Needed	Ad-hoc	Structured	Documented	Standards
	4	Acquisition Focus	Service-Centric	Threat-Based	Simulation-Based	Capability-Based
	5	Type of System Architecture	Heterogeneous	Homogeneous		
	6	Terminology Maturity	Developing	Stabilized	Commonplace	Other
	7	Methodology Focus	Qualitative	Quantitative	Mixture	None
	8	Proof-of-Concept Capability Focus	Persistent Precision Engagement	Long Range Strike	Prompt Global Strike	Cooperative Airspace OPS
			Operationally Responsive Space Access	Persistent ISR	Multi-Mission Mobility	Other
	9	Primary Domain of Interest	Air	Land	Sea	Near-Space
			Space	Cyber	Other	
	10	Level of Heterogeneity	None	Low	Medium	High

Figure 20: Matrix of Alternatives for Observations and Assertions.

In this example, the matrix of alternatives lists key decisions required to focus the problem as the rows of the matrix. The columns of the matrix identify potential choices to address each decision. While at first it appears that there are only four alternatives for the problem definition phase, the actual number of combinations is defined by the items in each row multiplied by the items in every other row (assuming each decision is independent of other decisions). This innocent looking matrix has 1,572,864 possible combinations: with each variation defining a different research plan. The green highlighted elements in Figure 20 identifies the attributes chosen to scope the problem based on the observations made in the previous sections.

[12] See Section C.5.1 for an overview and history of this technique.

2.5.1 Modeling and Simulation Enables Quantitative Technology Evaluation

> *"Only the most naive scientist believes that the perfect model is the one that perfectly represents reality. Such a model would have the same drawbacks as a map as large and detailed as the city it represents... its specificity would defeat its purpose: to generalize and abstract."*
>
> -James Gleick
>
> *Chaos* [168]

Section 1.2 notes that most techniques for resource allocation are based on qualitative information and subjective analysis. For example, the Technology Development Approach (TDA) evaluates system-of-systems level Measures of Effectiveness (MoEs) using a committee approach [129]. Unfortunately, a committee approach is only valid when the physics of the problem are well understood and seldom extends to the system-of-systems level where the complex *interactions* between heterogeneous elements do not follow intuitive or predictable patterns. Also, since the experience base of subject matter experts is bounded by tacit information based on known situations and scenarios, an expert-driven process is often not appropriate to produce quantitative estimates of system effectiveness.

One method for quantitative evaluation would be to build and test the systems in question; however, for a military system architecture, such exploration would be cost prohibitive. An alternative that balances cost and fidelity is to use *modeling and simulation* as an enabler for quantitative analysis. **Based on the aforementioned observations, modeling and simulation is a necessary component of a methodology for capability-based technology evaluation for systems-of-systems.**

Modeling, "a simplified description of a complex entity or process," is literally the creation of a model [22]. Its complement is simulation, defined as "the process of imitating a real phenomenon with a set of mathematical formulas" [9] Simulation can also be described as the repeated exercise of a model under various conditions.

The use of modeling and simulation in the defense community is not new and has co-evolved dramatically with advancement in digital computers [308]. Since the introduction of

Integrated Product and Process Development (IPPD) in late 1993, Schrage has advocated a generic methodology that leverages a computer integrated environment to enable robust design simulation [34, 368]. This methodology "provides the means for conducting parallel process/product (cost/performance) design trades at various levels (system, component, part)" and enables "distributed design and development" [39].

According to the National Science Foundation, simulation "can be used to explore new theories and to design new experiments to test these theories" and "also provides a powerful alternative to the techniques of experimental science and observation when phenomena are not observable or when measurements are impractical or too expensive" [35]. The National Research Council notes that modeling, simulation, and analysis "is of value in the early stages of defense modernization, when roughly defined concepts can be examined and adjusted in virtual worlds" [310]. Furthermore, the DoD's Transformation Planning Guidance reaffirms a commitment to expanding M&S capabilities, noting "DoD must be able to support a capability-based planning process that accounts for greater uncertainty in threats and capabilities and must be capable of comparing risks across time and between multiple theater-level operations" [310].

While modeling and simulation is an enabling technique that provides a means to calculate MoEs for candidate technologies and system architectures, **many technical challenges arise from its use**. The technical challenges and resulting research questions are enumerated in Chapter 3. Hypotheses that recommend the infusion of techniques and methods from other fields are proposed in Section 3.4 to address each of these in turn.

2.6 Synthesizing a New Methodology: First Attempts

Based on the aforementioned observations, a new methodology to support capability-based technology evaluation is needed. To determine the necessary elements of this methodology, a functional analysis is used. A successful methodology must perform at least three functions:

- Outline a structured process (methodology)
- Establish analysis goals (capability-based)
- Analyze results (technology evaluation)

From this basic functional decomposition, several other required functions are derived.

- Define the control volume for analysis (scenarios)
- Build models to study phenomena (models)
- Produce results (execute simulation)

Defining the control volume and building models identifies what is to be studied, how it is to be studied, and under what conditions the study is performed. The production of results is directly related to the need to analyze results. These six elements form a "common sense process" that connects the six required functions into a baseline process as shown in Figure 21.

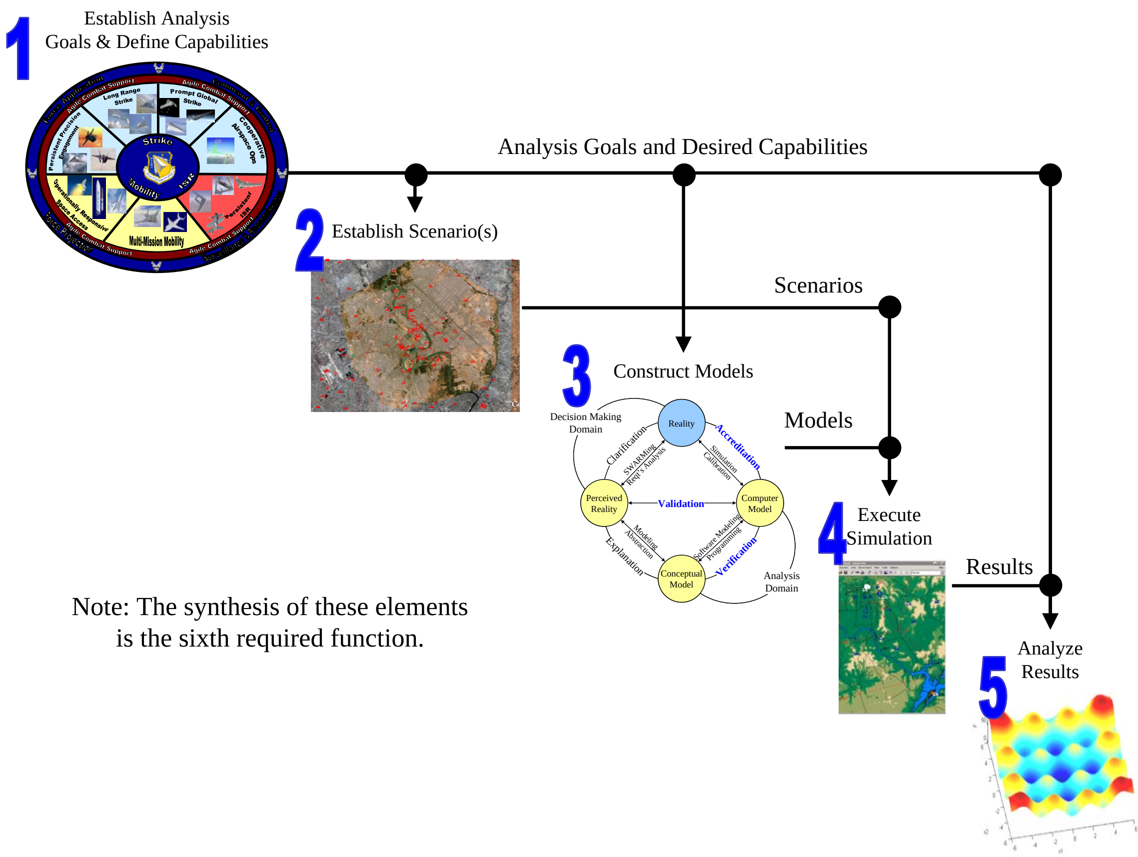


Figure 21: A Baseline "Common Sense Process" for Capability-Based Technology Evaluation.

While the process shown in Figure 21 has all the basic elements needed to analyze technologies, it suffers from several critical shortcomings. These are expanded as technical challenges in Section 3.1:

- The modeling and simulation architecture requires physics models (performance) and cognition models (actions).
- The mapping of top-level objectives to tactical execution is time consuming.
- Humans are often used in the analysis loop to make decisions about the next actions.
- Tactics are not variable as technologies are changing.
- Results analysis is difficult for large dimensionality problems.

Because of these challenges, the five step methodology shown in Figure 21 is not sufficient to solve this problem. As a result, a technical literature search was conducted to identify techniques and methods that can be synthesized to address a variety of technical challenges. These are summarized in Chapter III. The "common sense process" depicted in Figure 21 is supplemented with these enabling techniques to formulate a more complete methodology in Chapter IV.

CHAPTER III

TECHNICAL CHALLENGES, RESEARCH QUESTIONS, AND HYPOTHESES

A problem is "a question to be considered, solved, or answered" [22]. Chapter I outlined the nature of a problem facing the military acquisition community: the need to perform quantitative capability-based technology evaluation for systems of systems. A study or analysis applies a series of existing well-established techniques and best practices to the problem in an effort to discover a solution. Unfortunately, as summarized in the subsequent chapter, the best practices that apply to the identified problem are not well developed. No single "cookbook" for capability-based technology evaluation exists today.

In addition to the lack of a standardized methodology, the analysis of system architectures is also confounded by a number of technical challenges. A literature search was conducted across the technical domain to enumerate these challenges and identify mathematical techniques or methodical approaches that directly address critical shortcomings.

In this dissertation, a *new method* is proposed that leverages advances in modeling and simulation to enable quantitative technology evaluation with respect to top-level capabilities. While the selection of modeling and simulation as a technique provides the ability to calculate the change in capability-level MoEs as technologies are applied, the use of modeling and simulation is not without penalty.

3.1 Technical Challenges

While the National Science Foundation notes that "computer simulation is an indispensable tool for resolving a multitude of scientific and technological problems facing our country," [35] the implementation of computer simulations of complex systems is hindered by several technical challenges :

- The run time for military simulations is often extreme. Processes must be sped up to the point where running cases is trivial to allow effective design space exploration.

- A system-of-systems relies on many heterogeneous, interoperating assets to provide a capability. Each element requires a model of some kind. The problem space quickly becomes unmanageable.
- The models which must be accurately constructed to represent the phenomena under test with the appropriate degree of fidelity.
- Techniques are needed that reduce the scale of the simulation to the relevant phenomena while retaining the emergent behavior derived from system interactions.

In addition to the challenges that arise from the need to simulate results, some general technical challenges are also present:

- There is no structured process for performing quantitative technology assessments for systems-of-systems.
- The importance of individual elements and the sensitivity of the overall design environment to changes in fidelity at the asset level must be understood and accounted for.
- Uncertainty grows as multiple elements and interactions are implemented in the environment. This uncertainty must be quantified where appropriate.
- Capability-level MoEs are often difficult to define and may be scenario dependent.
- Military simulations often use a human-in-the-loop to make decisions at major junctions in time. It is not practical to have a human make thousands of decisions as cases are executed in bulk for technology forecasting.
- A variation in tactics or doctrine may have a much greater impact on top-level capabilities than a change to asset-level attributes. The relationship between tactics and technology must be managed.
- Optimization for a system-of-systems may be inappropriate as few systems-of-systems are employed according to point design conditions.

The proposed methodology must effectively use simulation to address the need for capability-based technology evaluation. Appendix C details several approaches to address the aforementioned shortcomings by cross-fertilizing techniques from other disciplines into the military modeling and simulation community. The current state-of-the-art is limited by these technical challenges: they act as roadblocks that prevent further realization of benefits due to simulation.

3.2 Research Questions

These technical challenges result from a detailed definition of the problem and many modeling and simulation activities have been limited by one or more of the above challenges. These challenges stimulate the development of research questions that must be answered to address the motivating problem. The research questions below seek to identify tools, techniques and methods to overcome one or more of the technical challenges in the previous section.

1. How can the impact of technologies infused at the system level be analyzed at the system-of-systems level and compared to measurable performance metrics related to capabilities?
2. How can military simulation runs be executed without a human in the loop to make strategic and tactical decisions?
3. Should the goal of the methodology be to identify an optimum technology portfolio that maximizes effectiveness or to seek a balanced portfolio that is robust across envisioned operating conditions?
4. How can the scale of the problem be appropriately reduced without losing the essence of the problem?
5. For a given problem, what is the best way to determine the necessary elements of a system architecture?
6. How can the simulation process be sped up to allow examination of the design space in a reasonable time frame?
7. What sampling techniques and modeling techniques are valid for non-linear systems-of-systems simulations?
8. How can the importance and sensitivity of individual elements (or degrees of freedom)

of the system architecture be evaluated?

9. How can uncertainty be quantified for this class of problems?

Capability-based technology evaluation has not been realized to date due to the technical challenges that serve as roadblocks in any previously proposed process. **The above research questions directly address the technical challenges**. While completely answering the set of questions removes the roadblocks and enables a solution to the problem, answering some questions reveals additional questions that were in the noise until the first "layer" of questions was answered. The analogy to modern air combat is direct: technical challenges are like air defenses that must be rolled back, revealing additional defensive layers. Each "threat" must be dealt with in turn. Research questions stimulate the development of courses of action to address the technical challenges. The proposed courses of action are *hypotheses*.

3.3 Hypothesis Genesis and Development

> *"A fact is a simple statement that everyone believes. It is innocent, unless found guilty. A hypothesis is a novel suggestion that no one wants to believe. It is guilty, until found effective."*
>
> -Edward Teller

A hypothesis is "a statement of conjecture subject to proof" [188] and is derived from the Greek, *hypotithenai* meaning "to put under" or "to suppose" [16]. Although these suppositions are critical to the development of scientific theories, according to Polya there is no universal or logical process for the development of new ideas. They are only developed by creative intuition. The experience base from which intuition results is contingent on the beliefs of the examiner [343]. Hypotheses can be proposed based on commonly held beliefs or myths. They can also be reasoned by analogy to a similar, yet well understood phenomenon.

One of the concerns about hypothesis formulation is that formal hypotheses are too constrictive and thus impede serendipitous discovery; however, scientists should always be on the lookout for non-intuitive results. Awareness of the possibility of unintended outcomes is critical to the discovery of asymmetric solutions. Furthermore, Kass notes that another criticism of hypothesis development for warfighting experiments is that hypotheses "are supposed to be derived from theory and there is no military theory" [233]. He addresses this concern by advocating a thorough literature search prior to hypothesis formulation and says "few, even science experiments, are derived from formal scientific theories" [233].

Since Aristotle's time, scientific progress has relied on philosophical reasoning as a means of establishing a line of inquiry and observations to establish a system of beliefs. Central to the development of hypotheses is therefore a development of beliefs and observations through series of literature searches. The subsequent section identifies and categorizes a set of hypotheses proposed to address the research questions previously stated.

3.4 Hypotheses Taxonomy and Exposition

Continuing the analogy between warfare and research, a hierarchy of hypotheses can be generated which addresses the complexity and method for answering the research questions.

20th Century air power theorist John Warden defines the levels of war as follows [474]:

- **Grand Strategic:** "where the most basic but most consequential decisions are made." Hypotheses at this level are philosophical and directly address the problem statement in the motivation. The primary research objective is stated at the Grand Strategic level.
- **Strategic:** "concerns the overall conduct." Philosophical arguments that fundamentally shape the research activity are appropriate at this level. Strategic aims shape the overall approach and derive lower-level research questions.
- **Operational:** "is primarily concerned with how to achieve the strategic ends." General methodology-specific approaches are formulated at this level.
- **Tactical:** where "objectives are unambiguous." Though most frequent in number, hypotheses at the tactical level are primarily implementation related. Selection of techniques at the tactical level is primarily conducted through a literature search of best practices. Just as individual "mistakes" at the tactical level in war rarely turn the

tide in combat, selecting the “wrong” techniques at the tactical level seldom impact the strategic aims of the methodology.

The hypotheses are summarized in Section 3.4.5; however, first it is necessary to summarize the results of a search of the technical literature to formulate the hypotheses. The specifics of the literature search are given in Appendices A, B, and C.

3.4.1 Grand Strategic Hypothesis: Research Objective

The single grand strategic hypothesis is the overarching research objective that comes primarily from the motivation section of this dissertation and directly addresses a top-level need. While the proposed methodology must be consistent with this objective, the research plan outlined in subsequent chapters may not be a panacea that complete addresses the community's need due to the large scope of the motivating problem. Simplifications are made where appropriate to facilitate methodology development. As a result, generalization of the single test case (application to a Long Range Strike system architecture) to the domain of all possible problems is inappropriate.

Hypothesis: The focus of this research is on the development of a valid, defensible, and practical methodology that facilitates a quantitative assessment of technology potential of systems-of-systems with respect to capability-level gaps and provides information to decision-makers early in the design process.

3.4.2 Strategic Hypotheses

Strategic hypotheses must be consistent with the research objective and *are more philosophical in nature.* They shape the definition of lower level hypotheses and define the basic structure of the proposed methodology. Exposition of the hypotheses defined herein can be found in Section A.

In Section 2.5.1, simulation was identified as a key enabler to facilitate quantitative analysis of technologies. The hierarchical structure and heterogeneous nature of military system architectures demands that the simulation be holistic because the interactions between system elements make it extremely difficult to analyze elements of the system in isolation and integrate the results *a posteriori*. Furthermore, constructive simulation (which involves simulated people in a simulated world) is most appropriate for analysis in which large numbers of cases must be run. Finally, an object-oriented architecture is appropriate when instantiating multiple elements with similar properties. Interfaces are generally more well-defined than hard-coded simulations and reusability and flexibility are enhanced. Object-oriented simulations are the standard in the military simulation community [333]. Section A.1.7 reviews existing simulation tools and selects one for this work.

With rare exceptions, wargames are generally not used for technology evaluation but for operational analysis against projected threats [87, 167, 187]. These monolithic simulations usually take days or weeks to execute and require a large staff to support and execute the simulation [196]. Since technology evaluation requires a thorough design space exploration consisting of many parametric runs as opposed to a single point design, the current mode of operation is not appropriate for this research. As a result, techniques to relieve the burden of human-based tactical decision making are needed to facilitate large-scale design studies. Machine learning and agent-based modeling techniques lend themselves to this application.

According to Anderson, Campbell, and Chapman, "analyzing performance of several design options of a complex system-of-systems across external parameters and multiple MoEs can generate a massive number of trade space combinations to be assessed, presenting extreme computational issues" [44]. While Section 2.3.1 noted that complex systems are often dominated by several strong interactions that cannot be decomposed without losing the essence of the system, weakly coupled interactions that do not contribute significantly to the variability of the MoEs should be set to default values to reduce unnecessary complexity.

Most simulations have near infinite degrees of freedom: assumptions must be defined to constrain the problem to a geographic region, a force size, technology level, political state, and the like. Weather conditions, sea states, supply availability, morale, and component failure rates must often be set to default settings to reduce the degrees of freedom to a manageable set. Recently, an increasing body of work relating to the optimization of system-of-systems has been published [237, 482]. While technically an extension of existing optimization techniques to the system-of-systems level, optimizing a system-of-systems for a single design point that may never occur may not be a valid approach to systems-of-systems design. Due to the sensitivity of large-scale complex systems to initial conditions, assumptions, and inflection points (or turning points in military terms), the concept of a robust technology portfolio is more appropriate than an optimum design.

3.4.3 Operational Hypotheses

If strategy is "the art and science of developing and employing instruments ... to achieve objectives" then operational hypotheses should focus on the mechanics of how these instruments may be comprised [468]. **The primary focus of new methods application is directed at the operational hypotheses**[1].

First, a technique is needed to prioritize targets within the simulation in a similar manner to the Master Attack Plan created by campaign planning experts [177]. A simple and rapid technique for prioritizing alternatives is through the use of an overall evaluation criterion. This multiple-attribute decision making technique relies on preference weightings. These weightings can either be directly specified or derived from another source. In this work, the Quality Function Deployment (QFD) technique is coupled with an Overall Evaluation Criterion (OEC) to decompose top-level strategic objectives into actionable tactical actions that can be implemented at the agent level. The specifics of this approach are detailed in Section B.1.

Next, a migration from a single point design to a design space exploration that examines cases in bulk to develop trends for architecture technologies requires a number of runs through a suitable simulation environment. Using a human to make every minute decision in the environment is not reasonable. After reviewing potential approaches (see Section B.2), a proposed approach is to use machine learning coupled with surrogate models to create an intelligent battle manager or "Meta-General" that has a rudimentary understanding of basic

[1] Because strategic hypotheses are mostly philosophical in nature, they must be generally accepted to justify research in an area. Tactical hypotheses are primarily implementation related and answer only esoteric research questions. As in warfare, the operational level provides a means to link tactics and strategy.

strategic decisions. The basis for this technique is well established in the field of computer science; however, it has not been applied to technology forecasting to date [3, 133, 375, 487].

Finally, one of the key problems in technology evaluation is the confounding impact of tactics on the operation of systems in the simulation. While the simplest approach uses hard-coded rules and assumptions to define agent performance, this approach is not scalable as new technologies are examined: each technology requires a new rule set. To overcome this difficulty, a more flexible set of rules and conditions can be used to formulate a "playbook" of tactical options. If the agent can gather enough information about its environment and its own performance within the environment, it can select the appropriate "play" to maximize its effectiveness. By using surrogate models of the performance space, an agent can explore the trade space of potential future decisions before those decisions are encountered. While chess-playing supercomputers attempt to emulate intelligence by examining an extremely large number of potential moves, an elegant approach that minimizes computational resources is preferred [209].

3.4.4 Tactical Hypotheses

Tactical hypotheses are focused on the application level and are derived directly from decisions made at higher levels. For example, three tactical hypotheses result from the first strategic hypothesis which advocates the use of an object-oriented constructive simulation to quantitatively assess technologies with respect to measurable capability-based performance metrics.

To begin, a simulation is comprised of models. These models can be simple linear trends, tables, empirical or semi-empirical models based on historical data, or models based on the laws of physics and natural constraints. The latter is most appropriate for the analysis of unconventional systems for which there is little historical data.

Once a simulation has been created within a framework and the simulation has been populated with models, how can technologies be simulated? For most technologies, a phenomenological simulation of the relevant physical effects is seldom available at the early stages of the design process, in fact, most of these phenomenological engineering simulation tools are written long after a technology has become commonplace and data exists to use for tool validation. A useful formulation developed by Mavris, Mantis, and Kirby uses "k-factors" as scale parameters on discipline level metrics[2] [293]. This allows existing simulation tools to emulate potential future technologies.

A technique called the Unified Tradeoff Environment (UTE) results from coupling physics-based models in a hierarchical modeling and simulation environment that uses k-factors across design variables, technologies, and assumptions [49]. This technique allows simultaneous trade studies across a variety of dimensions and enables migration from a point design to a parametric analysis of a variety of solutions; however, full realization of the UTE requires two other enabling techniques discussed below. The approach is also useful for assessing the relative sensitivity of a technology as assumptions are changed and uncertainties are applied.

The next series of tactical hypotheses relate to techniques used within the simulation.

A major challenge in systems-of-systems modeling is the large scale of the problem. Creating a simulation comprised of a variety of elements that must interoperate and pass information amongst each other in the form of messages is not unlike the issues faced by object-oriented software designers. The standard approach to software modeling is to use the Unified Modeling Language (UML) to formulate a storyboard of how code modules integrate together. An extension of the UML, the Systems Modeling Language (SysML) can provide a similar function when designing engineered systems. Additionally, the concept of the matrix of alternatives used throughout this dissertation to narrow the scope of the problem and to highlight method alternatives is traditionally used to delineate elements of a physical problem and the potential alternatives for subsystems or functions. The matrix of alternatives is a convenient way to identify the "threads" through the architecture that are the subject of a simulation.

Next, techniques are needed to identify what elements are included in the architecture to

[2] Originally termed "k" factors in the literature, the term gradually evolved to the more popular "k-factors" nomenclature subsequently used [240].

be modeled. One technique from systems engineering is the use of functional decomposition. According to the Defense Acquisition University, "concepts or designs are developed based on the functional descriptions that are the products of Functional Analysis and Allocation" [130]. A functional decomposition directly identifies the rows of the matrix of alternatives and brainstorming techniques and literature searches can be used to populate the columns with potential alternatives. Variations on this process are considered standard practice for systems engineers [217, 312]. In addition to functional decomposition, traditional brainstorming techniques and the SWARMing process advocated by Mavris (see Section C.6) are useful means to define system elements that satisfy the functions identified using functional decomposition.

While the previous techniques can address the technical challenges and are consistent with the grand strategic research objective, a key enabler is missing to make the proposed process a useful one. The concept of a surrogate model, a highly accurate approximation of a physics-based tool, is necessary to migrate from point designs to a *rapid* parametric tradeoff space. After a number of false starts and isolated applications at a variety of design organizations, Mavris successfully introduced surrogate models to the aerospace community in 1995. Since that time, they have become ubiquitous in the modeling of complex systems. The ability to reduce the speed of a process from minutes or hours to less than a second with no noticeable loss of fidelity enables the analysis of large volumes of data without supercomputing resources. While the integrated design environment previously mentioned is capable of analyzing technologies one point at a time, with minimal additional computational burden, a surrogate model can be created that enables a large design space exploration in a reasonable time frame.

While surrogate models based on a second order Taylor series expansion have become popular in the aerospace community, a major drawback of this approach is its inability to capture nonlinear or discontinuous behaviors endemic to military simulations. In the 1980's, researchers began experimenting heavily with artificial neural networks to specifically address this issue [195]. By the late 1990's, this technique had proliferated through the control systems community and was first applied to conceptual design [82, 116, 264]. Since

2005, the technique has become standard for any pattern matching process that cannot be approximated using a polynomial surrogate model: artificial neural networks are a key enabler for this research.

Surrogate models are created by developing a predictive capability that closely matches experimentally derived data, but models are most accurate when data is gathered in a structured manner. The concept of a "design of experiments" (DOE) is "a systematic, rigorous approach to engineering problem-solving that applies principles and techniques at the data collection stage so as to ensure the generation of valid, defensible, and supportable engineering conclusions" [318]. Put another way, allocation of experimental runs has shifted from "costly and time-consuming trial-and-error searches to powerful, elegant, and cost-effective statistical methods" [366]. While a variety of experimental designs have been created (and advocated) through the years to solve a variety of problems, an experimental comparison between several popular techniques was conducted (see Section 5.5.7). This experiment revealed that for this class of problems, a space-filling design using a sphere-packing scheme is the most effective technique for generating neural network surrogate models with the minimum expenditure of computational resources.

Since the strategic hypotheses identified the primary objective for capability-based technology evaluation to be robustness as opposed to an optimum point, a method for assessing the relative importance of input variables to the MoEs is needed. The prediction profiler, developed by Bradley Jones of the SAS Institute, is a graphical method for viewing partial derivatives as a function of all design parameters simultaneously. The ANOVA technique statistically assesses the relative contribution of a number of inputs on a response across a user-defined range of input values. Both of these techniques can be used to identify the significant parameters with respect to capability-level MoEs.

Engineering processes have many sources of uncertainty. A standard technique for the quantification and evaluation of uncertainty is the use of Monte Carlo simulation. In addition to uncertainty analysis, this technique is also useful for design space exploration. A linked, hierarchical, physics-based system-of-systems simulation provides the ability to quantitatively examine point designs. Creating a surrogate model of this simulation environment allows these point designs to be generated very quickly. By adding Monte Carlo simulation to the surrogate-enabled environment, data can be examined in bulk. Instead of searching for a needle in a haystack by running a handful of singular cases, a sweep of parameters can be quickly executed and the results graphically analyzed to find multi-dimensional optima using the technique developed by Bandte [52].

3.4.5 Summary of Hypotheses

The previous section summarizes the proposed solutions to each research question. Further exposition of the potential options for each decision and the process by which the preferred techniques are selected are given in Appendices A, B, and C.

The hypotheses postulated for this dissertation are summarized as follows:

1. **Grand Strategic (Research Objective):**

   The focus of this research is on the development of a valid, defensible, and practical methodology that facilitates a quantitative assessment of technology potential with respect to capability-level gaps. The methodology must be consistent with the JCIDS approach and focus on delivering additional information to the earliest stages of the capability planning process to assist technology analysts and military decision makers.

2. **Strategic:**

   2.1 A top-down capability-based evaluation of technologies can be performed using a holistic, object-oriented, hierarchical constructive simulation of systems-of-systems.

   2.2 Techniques from machine learning and agent-based modeling can be leveraged to provide an intelligent battle manager with an "understanding" of basic strategic and tactical decisions.

   2.3 For systems-of-systems, determination of a technology portfolio that is robust to changing threats and variable operating conditions is more useful than a portfolio optimized for maximum effectiveness.

3. **Operational:**

    3.1 Quality Function Deployment (QFD) and Multi-Attribute Decision Making (MADM) techniques can be used to prioritize targets based on desired strategic objectives.

    3.2 An "intelligent battle manager" created using agent-based modeling, machine learning techniques, and surrogate models can remove the confounding effect of tactics on technology evaluation.

    3.3 At the agent level, intelligence can be provided by surrogate models and tunable cognition models that allow the agent to forecast future decisions based on the effect of technology on system-level measures of performance.

4. **Tactical:**

    4.1 Physics-based models, implemented across hierarchical levels, are most effective because they can be mathematically validated.

    4.2 The concept of k-factors has emerged as a useful method for mapping technology impacts to surrogate model inputs.

    4.3 A hierarchical, integrated tradeoff environment is needed to analyze metrics at multiple levels.

    4.4 The inverse design technique is useful for setting targets at the top level and tracing these targets to measurable system attributes and technology performance metrics.

    4.5 A matrix of alternatives is a useful technique for reducing the scale of the problem using a defensible downselection process. The Systems Modeling Language is useful for diagramming code and programming modules.

    4.6 SWARMing and brainstorming were identified as techniques that are useful for scoping the problem and defining the appropriate "control volume" to be analyzed using modeling and simulation. A functional decomposition is a systems engineering technique that is useful in relating operational activities to system functions.

    4.7 Surrogate modeling is a proven approach that balances speed and accuracy to enable rapid design space exploration.

4.8 Neural Network surrogate models are appropriate for this class of problems because they can capture discontinuities endemic to systems-of-systems.

4.9 For non-linear systems-of-systems problems a space-filling design provides excellent coverage of the design space and facilitates the creation of neural networks.

4.10 The prediction profiler (essentially a matrix of partial derivatives) or the ANOVA technique can be used to ascertain what variables are significant with respect to the capability-level metrics.

4.11 Monte Carlo simulations are a convenient and uncomplicated method for quantifying uncertainty that have been used successfully for this class of problems.

A matrix of alternatives for the formulation of a design methodology results from the above questions (Figure 22). The *RQ* column identifies which research question is addressed by each row. Note that a research question can be answered at more than one hierarchical level. The *Hyp* column identifies the hypothesis within which the research question is addressed. Each of the options in the matrix of alternatives is detailed in the appendices: Strategic hypotheses are detailed in Appendix A, Operational hypotheses are addressed in Appendix B, and Tactical hypotheses are expounded upon in Appendix C. The orange shaded areas indicate options selected to address the research questions/hypotheses based on a review of the literature. Green shaded areas indicate selections that are justified by experiments performed in the course of this research.

What if different options are chosen in this matrix of alternatives? Figure 22 contains over 4.15 x 10^{22} options that could define variations on the proposed methodology. While changing the method options in Figure 22 modifies the proposed methodology, changes at the lower levels do not significantly impact the methodology's strategic aims. Over time, new techniques will be developed that can be added to this matrix of alternatives to potentially improve upon the methodology developed in this research or to apply it to different classes of problems.

	RQ	Hyp						
Strategic Postulates and Assumptions	1	2.1	Hierarchical System-of-Systems	Yes	No			
	1	2.1	Level of Heterogeneity	None	Low	Medium	High	
	1	2.1	Type of Simulation	Live	Virtual	Constructive	Interactive	
	1	2.1	Programming Approach	Custom (hardcoded)	Object-Oriented			
	2	2.2	Make Decisions in Simulation	Human-in-the-Loop	Decision Tree	Computer Assisted	Artificial Intelligence	Other
	3	2.3	Methodology Focus	Optimization	Robustness	Other		
Operational Hypotheses	2	3.1	Prioritize Targets	Random	Experience	QFD	OEC/MADM	Other
	2	3.2	Incorporate Tactics	Hold Constant	Include All	Use Static Mapping	Optimize for Each Technology	Other
	2	3.2	Type of Battle Manager Agent	Reactive	Deliberative	Mixture	Human	
	2	3.2	Battle Manager Learning Algorithm	Adaptive Neural Network	Genetic Algorithm	Bimodal: Training/Analysis	Random	None
	2	3.3	Type of Subordinate Agent	Reactive	Deliberative	Mixture	Human	
	2	3.3	Asset-Level Decision Rules	Hardcoded If-Then Statements	Random	Performance Vector of Attributes	Response Surface Equations	Other
	2	3.3	Asset-Level Learning Algorithm	Adaptive Neural Network	Genetic Algorithm	Bimodal: Training/Analysis	Random	None
Tactical Assertions	1	4.1	Type of Models	Physics-Based	Empirical	Hybrid	Other	
	1	4.2-3	Analyze Technologies	k-Factors	Unified Tradeoff Environment	Other		
	1	4.4	Trade Study Attributes	Point Design	Bottom-Up	Top-Down	Middle-Out	Other
	4	4.5	Reduce Scale of Problem	Committee Approach	SysML	Matrix of Alternatives	Other	None
	5	4.6	Determine Elements of Architecture	Provided by Customer	Functional Decomposition	SWARMing	Brainstorming Tools	Other
	6	4.7	Speed Up Processes	None	Linear Approximations	Qualitative Mapping	Surrogate Models	Other
	7	4.8	Type of Surrogate Models	Polynomial Response Surface	Neural Networks	Radial Basis Functions	Kriging	Other
	7	4.9	Sample from Design Space	Random	Full Factorial	Box-Behnken	D-Optimal	Uniform
				Orthog. Array	Space-Filling	Central Composite	Latin Hypercube	Other
	8	4.10	Evaluate Importance/Sensitivity	Committee Approach	Prediction Profiler	ANOVA/ MANOVA	Partitioning Techniques	Other
	9	4.11	Account for Uncertainty	Monte Carlo	Quasi-Monte Carlo	Petri Nets	Markov Chains	Other

Figure 22: Matrix of Alternatives for Strategic, Operational, and Tactical Research Questions.

CHAPTER IV

PROPOSED METHODOLOGY

The product of this dissertation is a methodology for capability-based technology evaluation. Section 2.6 outlined a generic method for military systems-of-systems analysis. This "common sense process" shown in Figure 21 is not realistic due to several key technical challenges enumerated in Chapter III. Following a literature search on enabling techniques and methods, a new holistic processes that addresses the observed shortcomings is proposed in this chapter.

4.1 Synthesizing a New Methodology: Refinement

After examining the functions that must be performed to quantitatively assess technology impacts, the baseline five step process in Figure 21 is extended to the ten step process shown in Figure 23. This process, which results from synthesis of enabling techniques and methods from Chapter III, is proposed as a methodology for capability-based technology evaluation. The process addresses the need for a structured approach to system analysis for large-scale systems-of-systems, and is validated through a proof-of-concept experiment based on a Long Range Strike system architecture in Chapter 5.

What are the appropriate measures of effectiveness to evaluate the ***process***? The most necessary condition is that the process works: by using the proposed process, engineers can *quantitatively* evaluate the impact of technologies across the system-of-systems hierarchy shown in Figure 10. The next measure of merit is that the methodology works *correctly*. By using the proposed method, do engineers get the "right" answers, or are the answers overly confounded by the large number of degrees of freedom? Another measure of merit is the speed with which the analysis can be conducted. While all processes generally add time to the analysis, the standardization of a structured method should add value by providing traceability and justification for resource allocation decisions. The development of a process

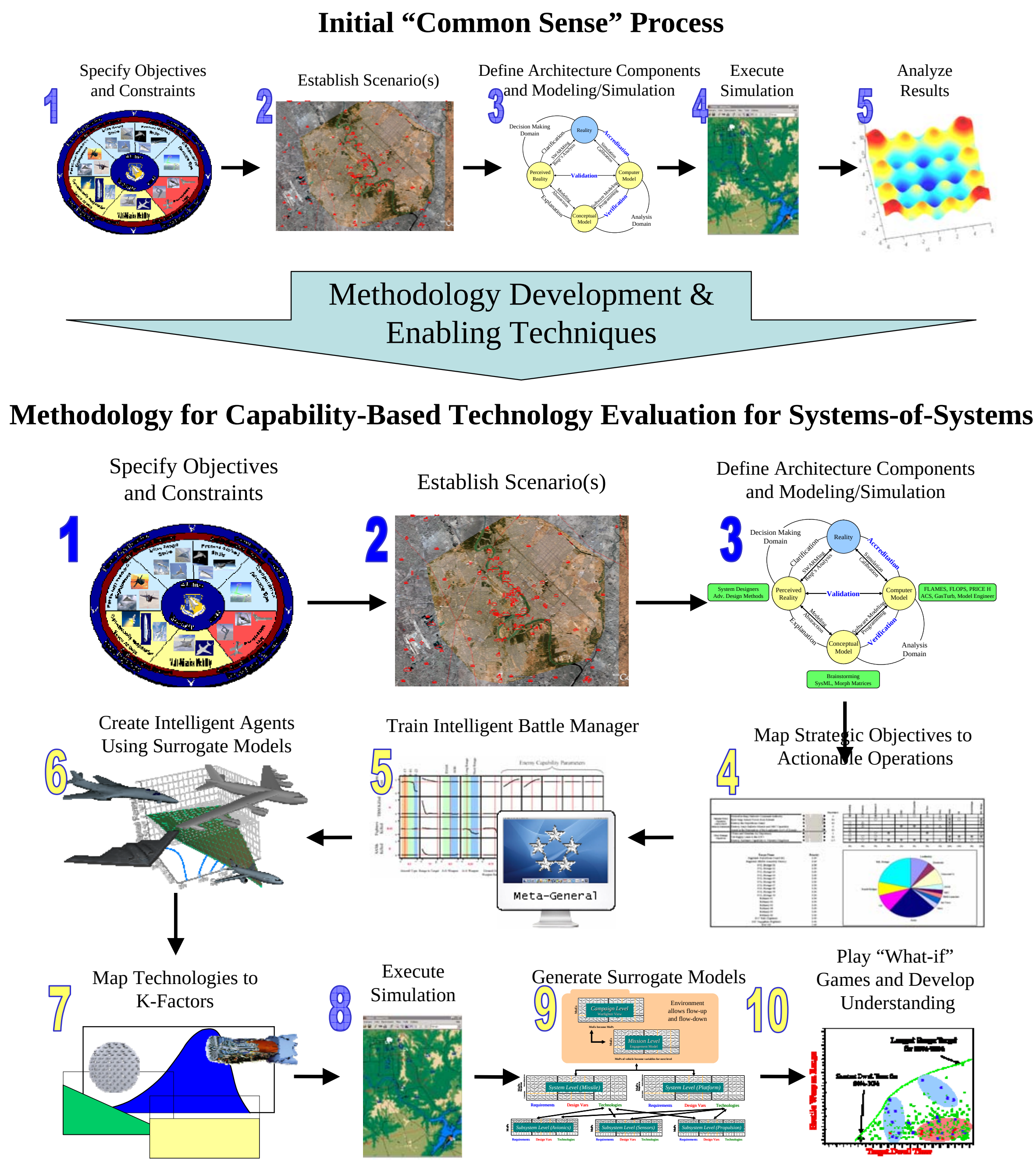


Figure 23: Evolution of a Ten Step Process Proposed to Address the Research Objective.

just to say "there is a process" is not a valuable activity. Finally, does the proposed methodology provide designers with information they do not already have, or does it merely restate the obvious? Can designers use the proposed methodology to make design decisions and validate those decisions against real-world models, or is the proposed method merely a curiosity?

These questions are revisited in the summary and conclusion to evaluate how well the proposed methodology satisfies these measures of effectiveness *using the LRS system architecture as a testbed example.* Also, a comparison is made to existing methods for technology evaluation using the metrics outlined in Section 1.2.8. The key aspects of the process shown in Figure 23 are described below.

4.2 Setting Up the Simulation Environment: Steps 1-3

Step 1 in a top-down, capability-focused technology evaluation is the establishment of the objectives of the technology evaluation study. This involves identifying capabilities, specifying the MoEs to evaluate capability gaps, constraining the simulation parameters, defining the timescale of the simulation, identifying models and tools that can calculate the needed MoEs, and selecting a simulation framework to perform the study.

Once the requisite capability or capabilities to be evaluated and the corresponding MoEs have been defined, **Step 2** is to create a relevant scenario or scenarios that exercise the system architecture in the appropriate operational environment. A technique called the "Estimate of the Situation" (see Section 5.2.1) provides a useful means to define the "who, what, where, when, how, and why" aspects of the scenarios used to evaluate a capability. The performance of a given architecture may be dependent on the conditions of the scenario: a robust technology portfolio for a wide range of military situations can be discovered by using the proposed methodology for multiple capabilities and multiple scenarios.

Step 3 in the process is the identification of system architecture components and a modeling and simulation environment to exercise those components in the selected scenario(s). The identification of models, linkages between models, required model fidelity, operational constraints, etc. must be defined in this time-consuming step.

It is important to note that steps 1-3 are required for *any* quantitative evaluation of technologies, and different processes in different organization exist to describe how a simulation can be created. Currently, the more ad-hoc processes emerge following step 3: once a simulation has been created, what do we do with it? **Steps 4 through 6 are aimed at addressing the primary technical challenge of capability-based technology evaluation and are the primary focus of new methods development in this work**

4.3 Accounting for Technologies and Tactics: Steps 4-6

In **Step 4**, the Quality Function Deployment (QFD) technique is used to map top-level strategic objectives to actionable operations. In the testbed example, the "operation" to be performed is "strike ground targets" and the ranking methodology identifies the order in which targets are to be struck (see Section B.1 for a technical overview and Section 5.4.1 for the implementation). It is important to note that while step 2 identified a scenario for analysis, by changing the importance of strategic objectives in this step, variations from the baseline scenario can be examined.

Next, **Step 5** takes the actionable operations from step 4, the simulation environment from step 3, the scenarios from step 2, and the MoEs that define a capability from step 1 and trains an intelligent battle manager to make decisions about force structure, technology options and tactical considerations with respect to the information provided in the previous steps. The output of step 5 is a "Meta-General" that is capable of performing the targeting and weaponeering functions, that is, matching weapons/platforms to targets with respect to their physical and functional properties. This step is useful in decoupling the human from the analysis loop, which is one of the major technical challenges for simulation-based technology evaluation. Background research on the techniques involved in this step can be found in Sections A.2 and B.2 while the proof-of-concept implementation is detailed in Section 5.5.

Step 6 addresses the need for centralized control and decentralized execution. Once assets (or agents) have been given technologies, what do they do with them? Given a variety of options, humans use a reasoning process to determine the "best" option based on their knowledge about each option and a prediction of possible outcomes. Intelligent agents can be used to simulate this process. In this case, information about each alternative comes from sensors that determine the attributes of the environment surrounding the agent. Since the agent has no intuition about which options are best, it must evaluate its predicted performance in each option and choose the option with the highest probability of success. While the human neocortex allows this process to happen almost intuitively (we seldom think about how we think), surrogate models can be used to enable the agent to forecast its performance under changing conditions. By creating a surrogate model that calculates MoPs as a function of discipline level metrics such as TSFC, drag coefficient, and weight, an agent can forecast how the MoPs change as technologies are implied. In this manner, an individual agent can forecast potential future actions and optimize tactical choices to maximize its probability of success with respect to information provided in steps 1-5.

In steps 5 and 6, surrogate models are used *inside* the simulation to provide tunable cognition at the battle manager and platform level. These surrogates are created *a priori* and inserted inside the code for the individual modules. This is in contrast to the surrogates used in step 9 that are created *around* the entire simulation and are generated after analysis cases have been executed.

4.4 Evaluating Technologies: Steps 7-10

In **Step 7**, specific technologies are identified for evaluation. Technologies can be a mix of specific system and subsystem level technologies or "architecture technologies" that represent a fundamentally different way of employing air power. The technology trade space must be defined in terms of k-factors that represent the impact of the technology on the system or subsystem-level metrics and the trade space must be bounded by realistic ranges. Furthermore, models must have been defined in step 3 that support evaluation of the technologies in question. For this reason, the proposed methodology may be an iterative process: as the technology space is better understood through modeling and simulation, additional models may be developed or fidelity may be increased to focus the technology evaluation study in different regions of interest.

This step facilitates the optimization of tactics with technologies because cognition models for the agents from step 6 can be "tuned" to alter the behavior of the agents within the simulation: the variation in the k-factors triggers different behaviors based on the outcome of the surrogate models. The purpose of step 6 is to identify *how* the agents should react to technology changes and the purpose of step 7 is to provide the agents with specific technologies to react to if a gap analysis is of interest.

Step 8 is where the computational effort occurs. After the simulation has been tested for execution errors, a design of experiments (DOE) across the technology space should be set up and executed in the simulation framework identified in step 3 using the scenario from step 2 with the adaptable agents from steps 4-6. This step is very computationally intensive and the end result is a table of data that represents the effectiveness of various technologies across the scenario(s) identified in step 2 for the capabilities defined in step 1.

Step 9 creates surrogate models of the simulation model that was executed in step 8. In contrast to the surrogate models inserted inside individual elements of the simulation in steps 5 and 6, the surrogate models in step 9 are created *around* the entire simulation system. The purpose of these surrogate models is to allow rapid domain-spanning explorations of the technology and tactics trade space. The hierarchical structure of the surrogate models is used to evaluate technologies across the simulation hierarchy depicted in Figure 10.

The use of surrogate models around the simulation allows a designer to examine cases that were not included in the original DOE and also enables implementation of probabilistic techniques for design space exploration, uncertainty analysis and visualization.

Finally, **Step 10** is where knowledge is gained by the process and where conclusions can be drawn regarding the effectiveness of a technology or portfolio of technologies. Using the surrogate models from step 9, a series of "what-if?" games can be played to evaluate technologies under a variety of conditions. Depending on the level of detail provided in the previous steps, iteration to any previous step may need to be performed. If the identified scenario(s) do not sufficiently evaluate the candidate capabilities, additional scenarios may need to be implemented in the modeling and simulation environment. Furthermore, some technologies may have been left out of step 7 and that step may need to be revisited.

4.5 Summary of Proposed Methodology

The methodology proposed is **SOCRATES**: **S**imulation-based, **O**bject-oriented, **C**apability-Focused, **R**eal-Time **A**nalytical **T**echnology **E**valuation for **S**ystems-of-systems.

Figure 24: The Death of Socrates, Jacques-Louis David (1787).

Socrates, the Athenian philosopher and mentor to Plato who lived from ca. 470 B.C. to 399 B.C "was the first person to question everything and everyone" [16]. This acronym is chosen because the proposed methodology is Socratic in nature, that is, information is gained by asking a series of questions to sequentially increase understanding[1]. Finally, Socrates realized that true wisdom comes from knowing that one knows nothing. Although designers may possess a large amount of tacit information about a system, when systems-of-systems are analyzed, expert designers must acknowledge that the additional degrees of freedom can confound even the most seasoned analyst.

The acronym incorporates the key elements of the solution. Object-oriented simulation is the medium chosen to enable capability-focused real-time analysis of technologies for systems-of-systems. The proposed methodology is designed to address the research questions from Section 3.2. The methods and techniques utilized in this methodology are explained in detail in Appendices A through C. The steps in the proposed methodology are summarized in Figure 23 and the specific options to address the hierarchical research questions are summarized in Figure 22.

[1]The Socratic method, described by Plato in the *Socratic Dialogues*, uses a series of questions to determine the beliefs and knowledge of a person of group of people. Using this technique, progressive hypotheses are eliminated by identifying contradictions based on known evidence and inferences [16].

CHAPTER V

APPLICATION OF THE PROPOSED METHODOLOGY

Chapter III proposed a series of enablers in the form of hypotheses to answer a set of research questions and address technical barriers to quantitative technology evaluation and Chapter IV identified a ten step process that synthesizes these elements into a structured methodology. However, postulating a methodology for the analysis of system effectiveness does little to address the need without a means to validate the proposal: imagination and philosophy can only carry an argument so far.

In contrast to inductive proofs and reasoning by analogy, the scientific method provides a means to deduce objective truths about the world around us [243]. "An integral part of this method is the idea of controlled, repeatable experiments to test hypotheses about how the world can be the way it is. Without a laboratory in which to perform these kinds of experiments, there can be no such thing as a bona fide *scientific* theory of anything," says John L. Casti of the Santa Fe Institute. The lack of an experimental environment to generate quantitative evidence of the method's effectiveness is much "like the barrier faced by early protobiologists without microscopes" [168]. To use the scientific method as a means to validate the proposed methodology (essentially an unproven theory), an experimental environment that exercises the proposed methodology on an example problem (Long Range Strike) is required.

As mentioned in the Motivation chapter, computer simulation "provides a powerful alternative to the techniques of experimental science and observation when phenomena are not observable or when measurements are impractical or too expensive" [35]. In the spirit of the scientific method, this chapter validates the proposed methodology by constructing a computer-based simulation environment to test the predictions made in Chapters III and IV. The subsequent sections of this chapter describe how this "virtual laboratory" is used to test the proposition through a proof-of-concept exercise.

5.1 Step 1: Define Objectives and Constraints

The first step of the SOCRATES method is to define the goals, objectives and constraints for the technology evaluation activity. These factors must be defined with respect to the research objective:

- Must be consistent with the JCIDS process
- Identifies a robust portfolio of technologies that best provides one or more capabilities
- Integrates multiple heterogeneous elements that comprise a "system-of-systems"
- Assesses the effectiveness across a broad trade space encompassing multiple concepts, technologies, tactics, missions, and domains

The objectives listed above guide the identification of analysis questions and "what-if?" games to be played in Step 10. Although the identification of a robust portfolio of technologies would ideally be conducted across multiple scenarios and capabilities, the proof of concept exercise is limited to a single capability and a single scenario comprised of multiple missions. Both of these are detailed below.

5.1.1 Identify Capabilities to be Studied

Next, a capability or capabilities to be examined must be defined to establish the baseline against which a technology performance can be measured. As highlighted in Section 1.3, Long Range Strike capability is selected as a relevant problem of interest to demonstrate the proposed methodology. The definition of this capability defines the trade space of potential technology options and suggests MoEs used to measure the ability of a solution to provide a capability.

An initial list of candidate MoEs includes the duration of the conflict (hours), the attrition levels across all target sets (%), number of strikes flown, number of munitions fired, cost of munitions fired ($), number of TBMs fired, average time to target (hours), number of blue units lost, and percentage of time critical targets killed. A detailed definition of specific MoEs is described in Section 5.2.2 because the calculation method is dependent on the scenario definition. In practice, not all desired MoEs may be calculable. The Air

Force Task List defines several high-level MoEs for precision engagement and global attack including percentage of effects achieved, percent of successful engagements, time for the desired effect to be achieved, percent of Earth's surface area accessible to USAF strategic attack, and cost to perform attack; however, no common set of MoEs has been defined across all scenarios and missions [428].

5.1.2 Define Analysis Questions

The primary focus of quantitative technology evaluation is the identification of a portfolio of technologies that best provide the required capability. Section 3.4.3 noted the confounding impact of tactics on technology forecasting. Any analysis must enumerate or account for the relative contributions of technologies and tactics on the MoEs either directly or indirectly.

The "virtual library" created to demonstrate the validity of the SOCRATES methodology for technology evaluation is also used to demonstrate the *usefulness* of the technique by answering a series of "what-if?" questions that highlight the benefits of parametric quantitative analysis. Some analysis questions include:

- What is the composition of a robust portfolio of technologies that provides Long Range Strike capability?
- What is the relative contribution of tactics and technologies to the overall effectiveness of the proposed portfolio?
- How do different aspects of the system architecture such as sensors and munitions contribute to the effectiveness of a technology infused platform?
- Can new munitions be leveraged with existing platforms or are platform technology advancements required for an effective system?
- How does the selection of a technology portfolio depend on the evolution of enemy capability?
- How does the definition of a mission influence the composition of a technology portfolio?
- Can the flexibility of the object-oriented framework demonstrate the reusability of the simulation components for the analysis of multiple architectures?

The identification of analysis questions defines to some degree the selection of scenarios and requirements for analytical models as they must be developed to answer the analysis questions. Furthermore, the assumptions and constraints imposed on the scenario may also be defined by the specification of these analysis questions: a study on logistics and resupply may ignore the impact of terrain while the analysis of sensor systems would require precise geometric models of the targets to be analyzed, a detailed treatment of signatures, and high-fidelity terrain models for masking effects and line-of-sight calculations germane to the calculation of sensor effectiveness.

5.1.3 Define Models Needed to Address Analysis Questions

The development of the appropriate models is a time consuming process necessary to "set up the experiment" and is analogous to building a wind-tunnel model to the appropriate size and scale with pressure taps at the right location for the test in question. This step is detailed beginning in Section 5.3. It is important to note that models should not be implemented for the sake of implementing models. Since every calculation uses computer cycles that could be otherwise allocated to other models, the instantiation of every model should be traceable to the objectives of the simulation and the analysis questions identified for a particular study.

5.1.4 Define Timescale

It can be said that in war the first thing you lose is the plan. This is due to the Clausewitzian "friction" of war and the fact that the enemy adapts rapidly to a static plan [100]. Gordon and Trainor note that during Operation *Desert Storm*, Iraqi forces moved planes around every three days because they had determined it took at least three days for a nominated target to be written into the Air Tasking Order (ATO) [179]. To maintain consistency with the Global Strike Task Force's objective: "kick down the door into denied battlespace by rapidly degrading, and thereafter defeating, the adversary's C4ISR, anti-access weapons, WMD delivery systems, and threats to friendly ground or naval forces," a simulation time of three days is defined [249]. Beyond this time period, the logistical constraints of resupply and the adaptiveness of the adversary would need to be considered for an accurate simulation. Based on available resources, three eight-hour operational days were simulated for the longest of the operational missions defined. This assumption is predicated on the need for nighttime operations for current generation stealth aircraft. This also balances available computational resources with the size of the analysis DOE.

5.1.5 Constrain Unnecessary Degrees of Freedom

While constructive simulation can be used for a variety of analysis tasks including force mix determination, optimization, analysis of enemy defenses, and logistics analysis, the focus of the proposed methodology is on using simulation for quantitative technology evaluation. To assess the benefit of technologies on a system architecture, the following degrees of freedom are held constant:

- Systems types and numbers employed by both sides
- Logistics (resupply of ammunition and fuel)
- Command and Control algorithms and communications pathways
- Jamming technologies are not used
- All aircraft of the same type carry the same parametric munitions for each simulation run
- SEAD effectiveness is modeled through a scale factor on SAM density

This is not to say that these degrees of freedom cannot be altered; however, their variation confounds the issue of technology evaluation. Constraints are also important in bounding the problem within an acceptable control volume. The testbed activity focuses on mid-term LRS technologies that can be implemented within the 2018-2025 time frame based on estimates of LRS IOC date from the 2006 QDR [456]. While JCIDS seeks to develop integrated joint capabilities, the development of high fidelity models for joint force interoperability and network centric warfare are outside the scope of this research.

5.1.6 Select a Simulation Framework

In a physical experiment, a framework could be interpreted as the simulation apparatus, wind tunnel, or stress testing machine used to perform the experiment. In the case of simulation, a framework is the software used to execute simulations. While a variety of government and commercial simulation tools are available, a simulation framework should be object-oriented, able to be executed without a graphical interface, allow creation of additional models and interfaces, and should be unclassified and available for the simulation activity. A literature search of existing simulation tools was conducted in Section A.1.6. Comparisons based on the above criteria identified FLAMES by Ternion Software as the appropriate simulation framework for this work.

While the subsequent sections describe models developed in FLAMES, it is important to note that both the models and methodology are independent of the framework selected.

5.1.7 Determining the Relevant Experiments to Be Performed

As previously mentioned, the experimental setup (model development, linking, verification, and validation) for systems-of-systems is a complex and time consuming process that is not unlike devising a wind tunnel test to observe a phenomenon. In addition to the validation of individual models, interactions between multiple models and interactions with the experimental framework itself must be examined. The large scale of the modeling endeavor limits the scope of phenomenological investigation to a **single case**: analysis of a Long Range Strike architecture. This selection was made based on community interest, suitability of available models and tools for this problem, and the representative nature of the system architecture. While the steps of the methodology are not specific to the LRS example, future work is needed to validate the proposed process across multiple systems-of-systems before a more general statement regarding its universal applicability can be made.

5.2 Step 2: Establish Scenario(s)

"If there is one attitude more dangerous than to assume that a future war will be just like the last one, it is to imagine that it will be so utterly different that we can afford to ignore all the lessons of the last one."

-Former RAF Marshal, Sir John Slessor,
Air Power and Armies (1936)

The next step in the SOCRATES method is to establish one or more scenarios for analysis. The simulation uses the scenario as a "game board" on which to evaluate the MoEs that contribute to the capabilities defined in step 1. Scenario construction "establishes the context and boundaries of subsequent performance and technology evaluations" [187].

The dependence of the analysis process on the scenarios chosen poses several unique challenges. First, most scenarios of interest are sensitive or classified. Secondly, validation of a notional scenario is extremely difficult as there is no baseline for comparison. While several recent conflicts in Kosovo, Afghanistan, and Iraq have demonstrated the value of airpower, it is necessary to select a scenario for which some validation data exists. To test the proposed methodology, a scenario based on the 1991 Persian Gulf War (Operation *Desert Storm*), has been selected due to the large amount of data on this conflict available in the public domain[1].

The conflict between the United Nations coalition and the Iraqi Regime to liberate occupied Kuwait was a 42 day campaign dominated by the judicious use of airpower. Recreating elements of the air campaign meets the needs for validation and will provide an unclassified scenario for analysis. It is important to note that the next sections do not describe an animation or a replay of events past, but rather build upon known information to create a relevant scenario for exploration of new technologies and tactics against threats that can be parametrically varied in the presence of real geographic constraints.

[1] A number of after-action reports on the Serbian (1995) and Kosovo (1999) conflict remain classified as of 2006. In-depth analysis of post-2001 conflicts in Afghanistan and Iraq has not been completed.

5.2.1 Estimate of the Situation for the Testbed Scenario

The initial assumptions can be defined, with some liberty to focus on the Long Range Strike elements of the air campaign, using the commander's "estimate of the situation" as advocated by Dupuy [138]. Based on a U.S. Army process, the estimate of the situation has five key elements (Figure 25). Each of these elements are defined in turn in the subsequent sections.

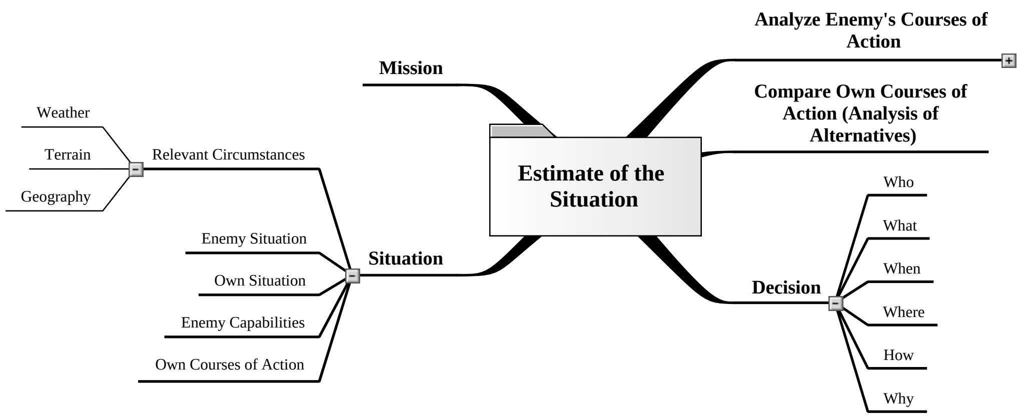


Figure 25: Estimate of the Situation (Adapted from Reference [138])

5.2.1.1 Mission

The stated goal of Operation *Desert Storm* was "to neutralize Iraqi National Command Authority, eject Iraqi armed forces from Kuwait, destroy the Republican Guard, destroy Iraqi's (sic) ballistic missile, nuclear, biological and chemical warfare capabilities as early as possible, and assist in the restoration of the legitimate government of Kuwait" [104].

To this end, the LRS mission is focused on the initial strategic air campaign, identified as Phase I by Glosson [177] and the first phase of the notional "Operation *Colorado Springs*" advocated by Dupuy [138]. In this phase, strategic targets including leadership, C4, and WMD facilities are aggressively targeted. Air bases and materiel production facilities may also be included on the prioritized target list. This phase was formally defined by the operational plan for the war as:

". . . attack Iraq's strategic air defenses, aircraft/airfields, strategic chemical, biological and nuclear capability; leadership targets; command and control systems; RGFC (Republican Guard Force Command) forces; telecommunications facilities; and key elements of the national infrastructure, such as critical Lots (lines of communications) between Baghdad and the [Kuwaiti Theater of Operations], electric grids, petroleum storage and military production facilities." [104]

Planners expected Phase I to last six to nine days [104]. Although strategic targets were attacked with air power throughout the conflict, air supremacy was declared after 10 days [457]. Given the time critical objectives of the Global Strike Task Force (GSTF), the simulated strategic air campaign simulated for the proof-of-concept activity is executed over at most six days. A more reasonable estimate based on GSTF operations in Operation *Iraqi Freedom* and Operation *Enduring Freedom* is between two and three days.

5.2.1.2 Situation and Relevant Circumstances: Overview of Iraq

Located in Western Asia, Iraq (Figure 26) has a land area 437,072 km^2, or roughly twice the size of Idaho. The geography for the simulation is bounded between 35° and 70° east longitude and 25° and 40° north latitude. Iraq is bordered by Iran (1,458 km), Jordan (181 km), Kuwait (240 km), Saudi Arabia (814 km), Syria (605 km), and Turkey (352 km) [91].

The population of 26,783,383 is largely confined to major population centers as shown in Figure 27, and nearly 50% of the country is uninhabited except for small nomadic groups [91]. A side effect of this population clustering is a relatively undeveloped transportation infrastructure confined to a few major highways linking the largest cities as shown in Figure 28. There are 45,550 km of roadways, 2,220 km of 1.435 m gauge rail lines, 5,418 km of oil pipelines, and three major ports at Al Basrah, Khawr az Zubayr, and Umm Qasr [91].

While the highest point in the country is at 3,611 meters elevation, the desert country is dominated by gently rolling hills and sparse vegetation [164]. A majority of the terrain is less than 400 meters above sea level. To further develop the scenario, the terrain of the region can be examined in detail. Digital Terrain Elevation Data (DTED), a matrix

Figure 26: Overview Map of Iraq [315]

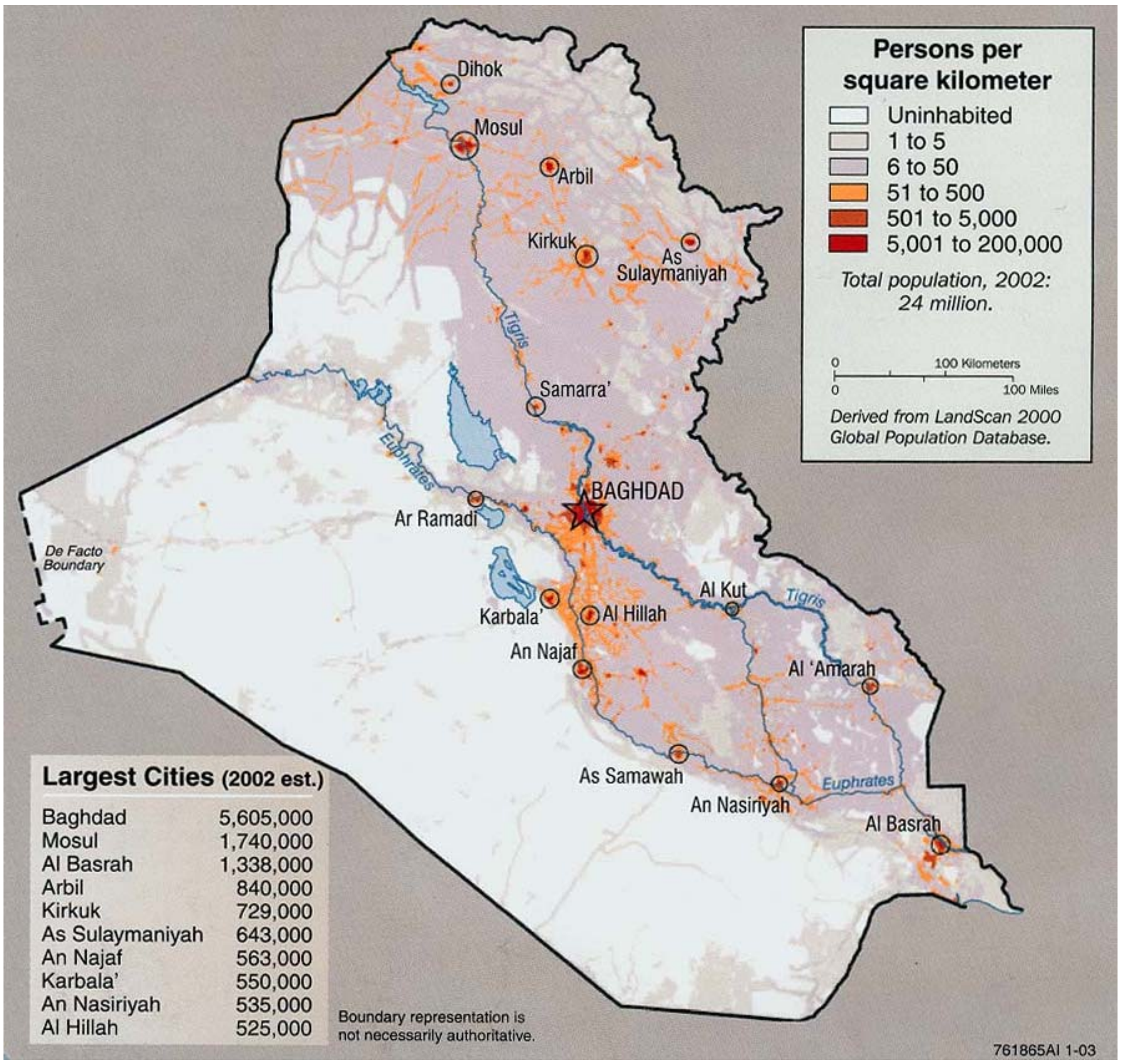


Figure 27: Population Density of Iraq [93]

Figure 28: Transportation Infrastructure of Iraq [93]

of terrain elevation values used for mapping and integration of geospatial information, has three levels of classification [150]:

- DTED Level 0: **Unclassified**, publicly available elevation information spaced at 30 arc seconds (one kilometer).
- DTED Level 1: **For Official Use Only**. Basic medium resolution elevation data used for military activities. This level is spaced at 3 arc seconds (approximately 100 meters).
- DTED Level 2: **Secret**. High resolution elevation data for military activities spaced at 1 arc second (30 meters). DTED level 2 was created using the results of the NASA STS-99 Shuttle Radar Topography Mission (SRTM) in February 2000 [313].

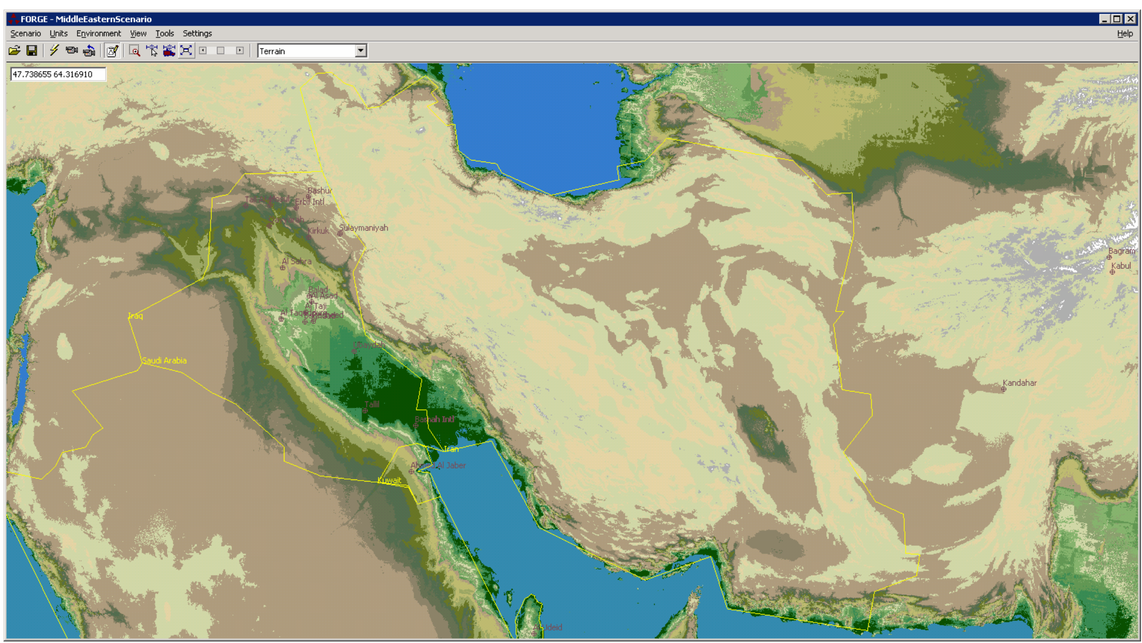


Figure 29: Baseline FLAMES Scenario Using DTED Level 0 Terrain Model.

High resolution DTED data can be used for line of sight calculations, navigation and targeting, route planning, and flight simulation [313]. Since none of these uses are planned with the current simulation, DTED level 0 is sufficient if terrain data is needed. DTED Level 0 for the region was obtained from the National Geospatial-Intelligence Agency (NGA) [316].

The geographic coordinates of countries in this region were calculated using Google Earth [33] and were superimposed on the FLAMES scenario as airspace regions (areas of responsibility). The baseline FLAMES scenario including the NGA terrain files, local airfields, and estimated geographic coordinates of local nations is shown in Figure 29. Loading the FLAMES dataset containing the terrain files significantly increases the time required to initialize the scenario. After examining a majority of the country for which operations are planned, the terrain features do not contribute significantly to the calculation of MoEs for this scenario.

The climate of Iraq is similar to that of southern Arizona, with summer highs reaching above 38° C (100° F) and winter lows below freezing. Precipitation is limited to less than about 18 cm per year, mainly between April and November. Dust and sand storms caused by wind patterns are the dominant weather phenomena, complicating operational issues for ground-based men and machinery as well as increasing the difficulty of aerial targeting. The desert has been described by World War II German General von Ravenstein as "a tactician's paradise and a quartermaster's hell" [138]. The lack of major terrain features provides near-infinite tactical mobility for ground units; however, high temperatures severely impact aircraft performance and pervasive sand and dust complicate the lubrication of machinery. As a result, aircraft takeoff field length, range, and loiter time are adversely impacted with respect to nominal conditions. In this harsh environment, sortie rates would be lower than a theoretical ideal due to increased maintenance demands.

While the weather of the region is generally clear, the period of January/February 1991 saw the worst weather in at least 14 years[2]. "Approximately 15 percent of scheduled aircraft attack sorties during the first 10 days were canceled because of poor visibility or low overcast sky conditions" [457]. Although pre-war estimates predicted cloud cover of 13% the average cloud cover was 39% over the duration of the conflict [208]. Low cloud ceilings (5,000 to 7,000 feet) impeded the ability to collect target imagery and perform battle damage assessment (BDA). Since no accurate weather model is available, *clear weather* is

[2]Although Reference [164] notes "conditions were no worse than what would probably be the best ones likely in other conflicts."

assumed for the short duration of the LRS mission; however, a scale factor for percentage of canceled sorties can be added to the environment to simulate negative effects of weather for future research activities. The FLAMES jamming model, while designed to simulate electromagnetic interference, can also be tuned to the visible spectrum, disrupting optical calculations if such detail is desired.

The 97% Muslim country has deeply rooted religious beliefs that restrict targeting around religious and historical sites. Additionally, military operations may be constrained by the Muslim holy month of Ramadan, which is based on the lunar calendar and hence migrates from year to year. In 1991, Ramadan fell between March 19^{th} and April 17^{th}. Constrained on the forward end by United Nations Resolution 678, military operations could begin no earlier than January 15^{th} of that year, limiting the entire scope of the campaign to 63 days [420].

5.2.1.3 Enemy Situation: Defining the Centers of Gravity and Locating Enemy Targets

> *Only by constantly seeking out the center of his power, by daring all to win all, will one really defeat the enemy.*
>
> -Carl Von Clausewitz,
> On War [100]

The *center of gravity*, a concept from mechanics[3] extended by Clausewitz to military theory is best defined as the "hub of all power and movement, on which everything depends" [100]. In contrast to absolute war[4] the purpose of targeting is to efficiently attack the enemy's center of gravity, ending military conflicts as quickly as possible with minimum cost and casualties. To accomplish this goal, it is first necessary to identify the elusive center of gravity which changes based on the condition of the scenario and the disposition of enemy forces.

[3]In mechanics, the center of gravity is the point through which the application of a force yields more work than the application of the same force through a different point.

[4]Absolute war is defined as the limitless expenditure of resources to utterly destroy the adversary. Clausewitz believed that absolute war was impossible because the constraints of politics and morality would define a point of mutual capitulation at which hostilities would cease [100].

In the 1991 Persian Gulf War, strategic targets were classified according to twelve target sets as described in Table 1 [457]:

Table 1: Apportionment of Targets Across the Twelve Target Sets [104, 450]

	Dec 1990	15-Jan-1991	Modeled
Leadership/Command	32	33	40
Electricity Production	16	17	10
Telecommunications and C3 Nodes	26	59	53
Strategic Integrated Air Defense System	28	101	126
WMD Research, Prod. and Storage	25	23	40
Missile Launchers, Production and Storage	N/A	43	54
Air Forces and Airfields	28	31	54
Naval Forces and Port Facilities	4	19	3
Oil Refining and Distribution	7	12	25
Railroads and Bridges	28	33	23
Ground-Based Army Units	N/A	37	1
Military Storage and Production	44	68	46
Total	**282**	**476**	**475**

Table 1 identifies (1) how these target sets were apportioned across the 282 targets identified by CENTCOM in December 1990 and (2) how the number of targets increased prior to the initiation of the air campaign on January 15, 1991 [104] and (3) how the targets were modeled in the experimental scenario[5]. Differences in the actual versus modeled values indicate the degree to which information is available on the targets struck during the conflict and encompass the fact that some targets belong to more than one target set.

Specific examples of facilities that fall into these target classes are listed in the Gulf War Air Power Survey [105]. The first five target sets represent 13% of strategic sorties (2,583) during the Gulf War while the sixth target set (mainly involving time critical theater ballistic missile (TBM) launchers) represents a surprising 15% of the strategic sorties (2,767) [118].

One of the properties of each target in the simulation is the target set(s) to which it belongs. While the first six target sets are traditionally priority targets for LRS assets, each target is prioritized by the battle manager for attack based on the relative importance of each type of target relative to the overall campaign goals. For example, if the primary goal

[5]Based on information in the public domain.

is an early halt to advancing ground forces, priority would be given to the destruction of air defenses, tactical communications systems, and the interdiction of fielded combat forces [119]. On the other hand, a "decapitation strike" aimed at quickly toppling a regime would utilize surprise attacks almost entirely centered on leadership and C3 targets. Nuclear, biological, and chemical (NBC) weapons production facilities would almost certainly be a priority target in any engagement scenario.

When considering the role of LRS assets to attack airfields and aircraft located on the ground, priorities for the first *seven* target sets are defined according to the guidelines set forth by Glosson and Brigadier General Larry Henry [177]:

- "Destroy the **leadership**, **NBC** targets, Republican Guard, **Air Force**, and Saddam's security forces, in that order."
- "Disrupt **C3**, industrial infrastructure, and other military facilities."

A method for parametrically varying the relative importance of each target set with respect to strategic goals is discussed in Section 5.4.1.

Initial developments in capability-based technology evaluation were focused on a set of identical targets uniformly distributed across a randomly shaped country [289, 61]. This simplifying assumption is a poor one due to an adversary's exploitation of the home field advantage provided by their geography. Analysis of the Iraqi region indicates that defenses were heavily clustered around a few population centers leaving large areas of the country undefended, as Cordesman notes, "Iraqi territory is too large to attempt territorial defense, and Iraq has always concentrated on defending strategic targets" [107]. Although this strategy is likely not the same for other countries with next-generation layered air defenses, detailed information on such arrangements is not available in the public domain. The representative setup used can be parametrically "tuned" to represent different capabilities within the same adversarial architecture. Using public domain sources, many of the high-value targets of interest enumerated by target set in Table 1 can be located:

- Generic regions of the country can be identified from map collections such as Reference [93] and Reference [466].
- Major targets in the Baghdad, region are defined by Davis [118]. The National Geospatial-Intelligence Agency [315] also provides public domain reference maps for Baghdad, Kirkuk, Basra, Tikrit, and Mosul.
- Additional strategic targets around Iraq are defined by Cheney [457], Cohen [104, 105, 106], Cordesman [107], and Putney [348].
- Airfields in the region can be identified from the Gulf War Airpower Survey [104], O'Malley's seminal 2001 report for the RAND corporation [330], and a more recent online database of worldwide airfields [48].
- While the GAO states that only 15% of fixed NBC facilities were known before the war [164] these targets can still be simulated using information identified by the CIA in Reference [92] and the United Nations Special Commission (UNSCOM) in Reference [421].
- The U.S. Air Order of Battle, the dispersion of air assets across the area of responsibility, is given by Glosson [177] and Cheney [457].
- Refueling tracks and AWACS flight paths were established in Saudi Arabia according to the Gulf War Air Power Survey [105].
- SAM coverage regions for the major population centers were also defined in the Gulf War Air Power Survey [105].
- The number of SAM sites in each region are given in the Conduct of the Persian Gulf War Final Report to Congress [457], which are mapped generically to the previously identified coverage regions. Defenses around Baghdad are primarily located around the outskirts of the city, "extending over the general Baghdad area, as far as 60 miles outside the city" [164]. Individual SAM fire units were located randomly within the range circles specified in the Gulf War Air Power Survey [105].
- Since their exact locations were unknown at the time, mobile TBM launchers are distributed randomly across western and southern Iraq in regions identified by Rostker [357] and Hallion [190].

- The position of U.S. naval units can be determined from References [457] and [106]. Since the focus of the simulation activity is on Air Force LRS assets, the exact order of battle of naval forces is not needed for this scenario.
- B-52G aircraft operating outside the area of responsibility launched thirty-five Air Launched Cruise Missiles (ALCM's) in the opening hours of the conflict [457]. These assets are positioned over the Mediterranean Sea.

Using this information, a resulting buildup of friendly forces and hostile targets in the FLAMES framework can be created and is shown in Figure 30 for the Iraqi region and Figure 31 for the Baghdad area. The red circles indicate the SAM coverage regions identified in Reference [105] and the black lines indicate command and control pathways between elements of the Iraqi air defense system. Different icons are used to identify factories, palaces, chemical, biological and nuclear facilities, oil-related targets, military storage sites, airbases, bridges, SAM sites, radars, and other targets of interest.

Finally, while the Iraq scenario serves as a baseline for the demonstration of the SOCRATES methodology, Hallion notes that "every developed nation has within it a remarkably similar number of key targets (about 500) and aiming points (about 3,000)" [190]. Parametric variation of the target characteristics, geography, and friendly asset properties and locations can simulate different scenarios for the analysis of the same capability, albeit at a higher computational cost.

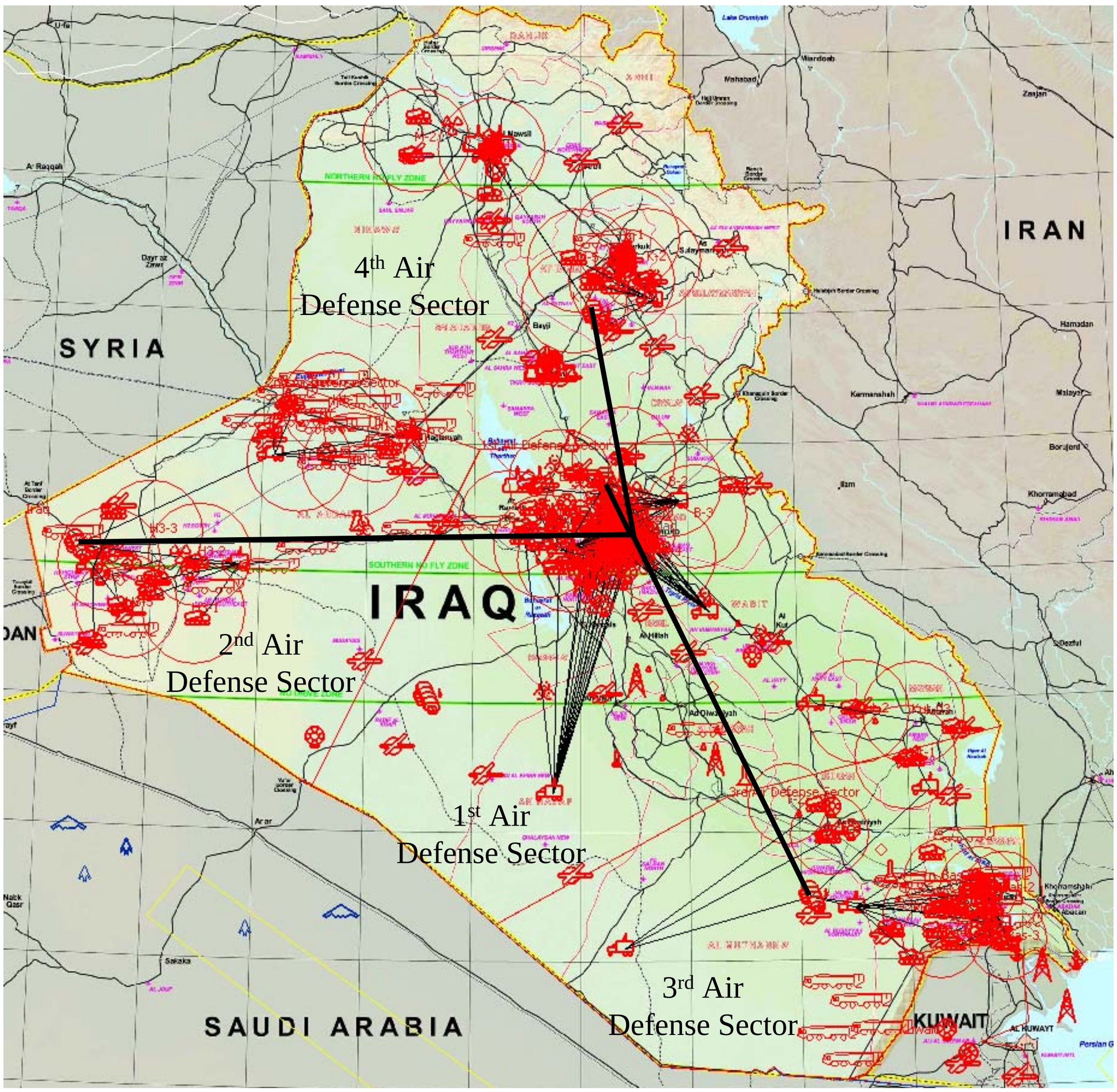


Figure 30: Iraq Scenario in FLAMES (Background Image from Reference [466]).

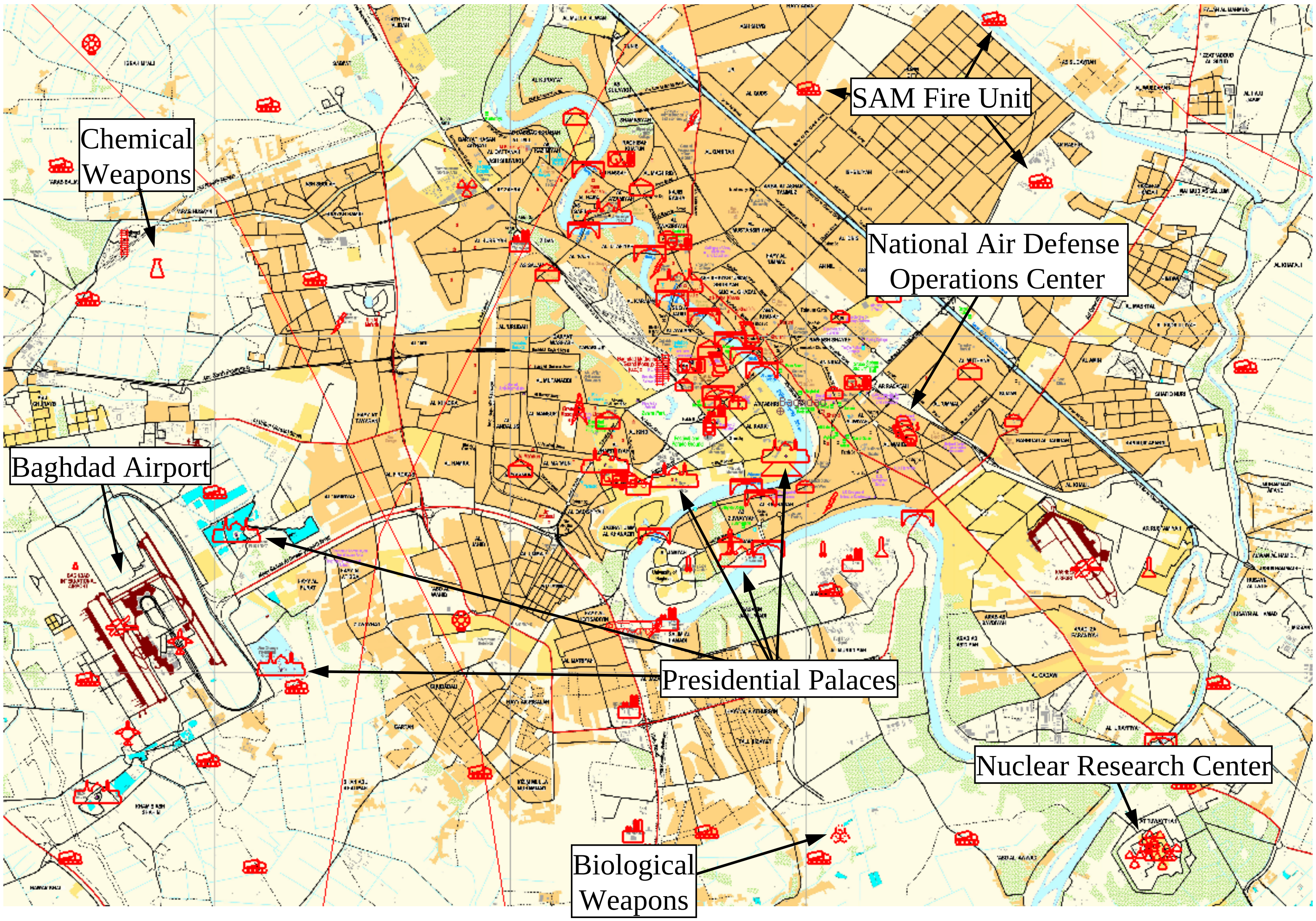


Figure 31: Target Laydown in Baghdad, Iraq (Background Image from Reference [315]).

5.2.1.4 Own Situation: Alliances and Order of Battle

Nearly 50 countries participated in the 1991 Gulf War, either through financial aid, troop commitment, or geographic basing. The states of the Gulf Cooperation Council (GCC)[6], Saudi Arabia, Bahrain, Qatar, and the UAE committed bases and logistics support to U.S. forces [457], with a majority of Air Force assets forward deployed in Saudia Arabia. Outside the main theater of operations, B-52G heavy bombers operated from RAF Fairford in the U.K., Moron airbase in Spain, and Diego Garcia in the Indian Ocean. Other major regional concerns which relate to the experimental setup include:

- Syria contributed special forces troops to the coalition, but its territory was not used for air operations.
- While Iran condemned the invasion of Kuwait, it also declared neutrality in the conflict.
- Lebanon did not play a role in the war.
- Egypt contributed ground troops, based in Saudi Arabia.
- The historical role of Jordan is not clear. The fairweather supporter of the Iraqi regime deployed troops to thwart a potential invasion of their largely undefended border. Jordan is considered neutral, but overflight is not permitted.
- Yemen and Sudan vocally supported Saddam Hussein.
- While 42 Scud missiles were launched at Israel, at the behest of America, the Israelis did not participate in the war[7].

The status of friendly (blue), neutral (white), and hostile (red) countries is shown in Figure 32. The basing situation of U.S aircraft are based on historical records from Reference [457].

The "Order of Battle" refers to the "number, type, and composition of forces available to a country or present at a battle" [106]. As shown in Table 2, the primary aircraft

[6] The GCC also includes Kuwait.

[7] While many historical accounts note that U.S. diplomatic efforts and defense with Patriot missile batteries kept Israel out of the war, a lesser known fact is that Israel requested, and was denied, access to Identification Friend or Foe (IFF) codes used by coalition forces [190].

Figure 32: Disposition of Neighboring Countries for the Simulation Activity.

used for bombing included the F-111F, the F-117A, the F-15E, and the B-52G[8]. Due to their relatively short range, F-16 aircraft were not utilized heavily in the strategic bombing campaign at the start of the war [177]. The Gulf War Air Power Survey also gives the unclassified basing information related to the initial deployment of forces at the 23 airbases in the area of responsibility [106]. Strike aircraft were generally based at Al Kharj (F-15E), Taif (F-111F), and Khamis Mushait (F-117A) [177]. Only the largest airfields in Saudi Arabia could support refueling aircraft and heavy bombers such as the B-52G, some of which flew from Moron, Diego Garcia, and Barksdale AFB, Louisiana. Several of the aircraft used are depicted in Figure 33.

[8]While the F-117A constituted less than 2.5% of the aircraft in theater, *Nighthawk* crews attacked 31% of the strategic targets on the first day of the war [424].

Table 2: Order of Battle for USAF Forces During the Persian Gulf War, January 1, 1991 [106].

Aircraft	Type	Number	Aircraft	Type	Number
F-15C	Fighter	96	E-3A	AWACS	7
F-4G	Electronic Combat	48	EF-111	Electronic Combat	18
F-16	Fighter/Attack	168	EC-135	Electronic Combat	0
A-10	Attack	120	KC-10	Refueling	6
AC-130	Gunship	4	KC-135Q	Refueling	164
F-117A	**Bomber**	36	C-20	Airlift	1
F-15E	**Bomber**	46	C-21	Airlift	8
F-111F	**Bomber**	64	C-29	Airlift	0
B-52G	**Bomber**	20	C-130	Airlift	96
TR-1A	Recon	2	HC-130	Special Ops	4
U-2	Recon	3	MC-130	Special Ops	4
RF-4C	Recon	6	MH-53	Special Ops	8
RC-135	Recon	4	MH-60	Special Ops	8
JSTARS	GMTI	2	EC-130E	ABCCC	6

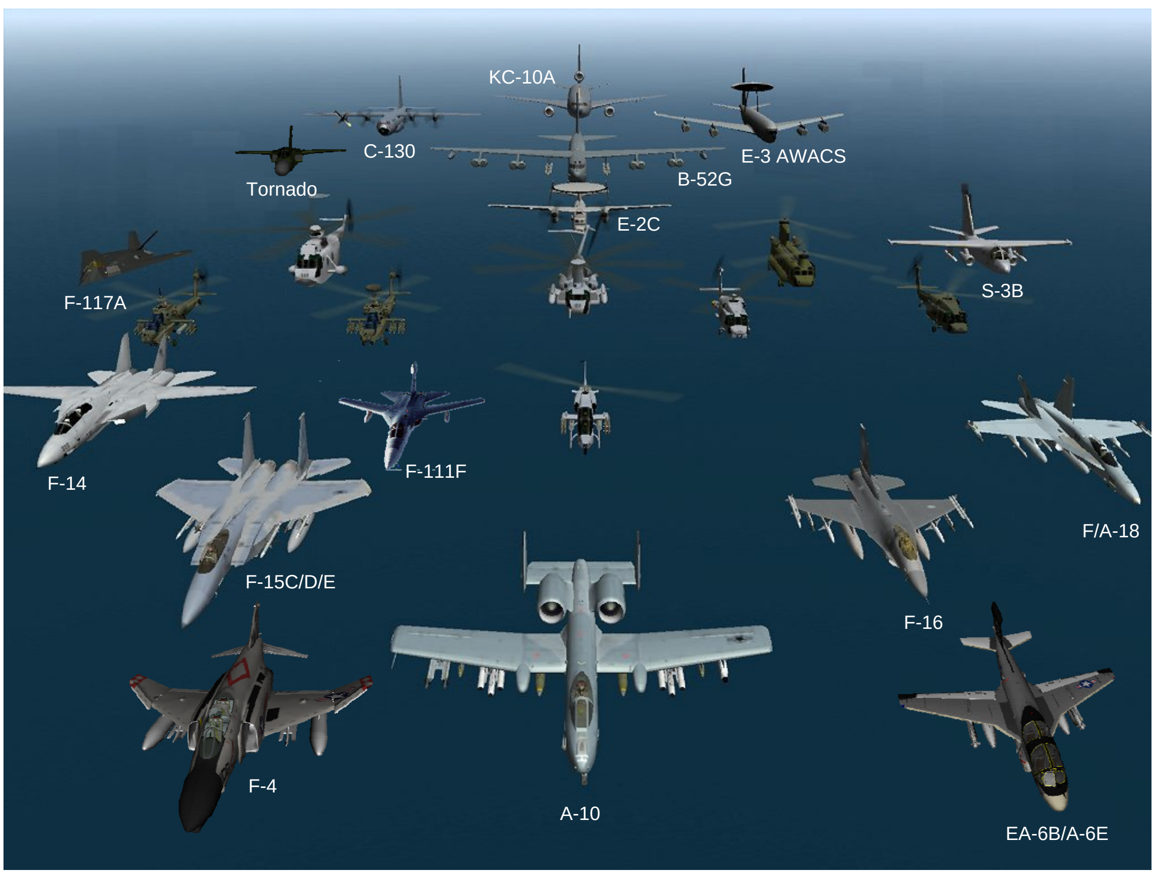


Figure 33: Aircraft Used in the Persian Gulf War (Adapted from Reference [417]).

Despite the recent reluctance of Saudi Arabia to support U.S. operations in the region [166], during the 1991 conflict, Saudi Arabia hosted a large percentage of U.S. strike assets. Support equipment such as jammers, SEAD aircraft, tankers, and short-range ground attack aircraft are based in Saudi Arabia according to the positions specified in Reference [457]. LRS assets are positioned at Khamis Mushait where F-117A aircraft were bedded down. Future work may examine other locations in Qatar, UAE, Oman, and the British island base of Diego Garcia to examine LRS effectiveness from multiple bases when Saudi facilities are denied.

5.2.1.5 Enemy Capabilities: Review of Air Defense Threats

At the outset of hostilities, Iraq boasted the fourth largest army, and the sixth largest air force in the world [190]. Protected by thousands of anti-aircraft systems, Baghdad was arguably more heavily defended than Moscow [404]. Using primarily Russian anti-aircraft equipment, a French-built Integrated Air Defense System (IADS) called KARI[9] linked Western and Soviet radar, SAMs, and anti-aircraft artillery (AAA) while providing redundant command and control. The KARI system was optimized to defend from the west (Israel) and the east (Iran), leaving a "dead zone pointed directly at Baghdad from Saudi Arabia" [105]. While the air defense system was primarily concentrated around Iraq's major population centers, a large number of high-value targets were located in and around these centers and along the Tigris and Euphrates rivers. Iraqi doctrine, which emphasized defense of the capital and these centers, enabled allied air operations within portions of the country to some degree. The centralized nature of KARI was both a strength and a weakness. On one hand, when integrated, the system was incredibly powerful; however, if the central nodes could be disables or destroyed, individual fire units would be forced into autonomous mode. Turning on their individual radars for target acquisition and tracking made them easy targets for U.S. radar-homing missiles.

Iraqi SAM air defenses were provided by the Soviet SA-2, SA-3, SA-6, and the Franco-German Roland systems [457]. Additionally, the 16,000 radar-guided and heat-seeking

[9]KARI is Iraq in French (IRAK), spelled backwards.

missiles included man-portable and other Iraqi army missiles such as the SA-7, SA-8, SA-9, SA-13, SA-14, and SA-16 [338]. Track information for KARI was provided by a network of early warning radars. The approximate SAM and Early Warning (EW) radar coverage of Iraq is shown in Figure 34.

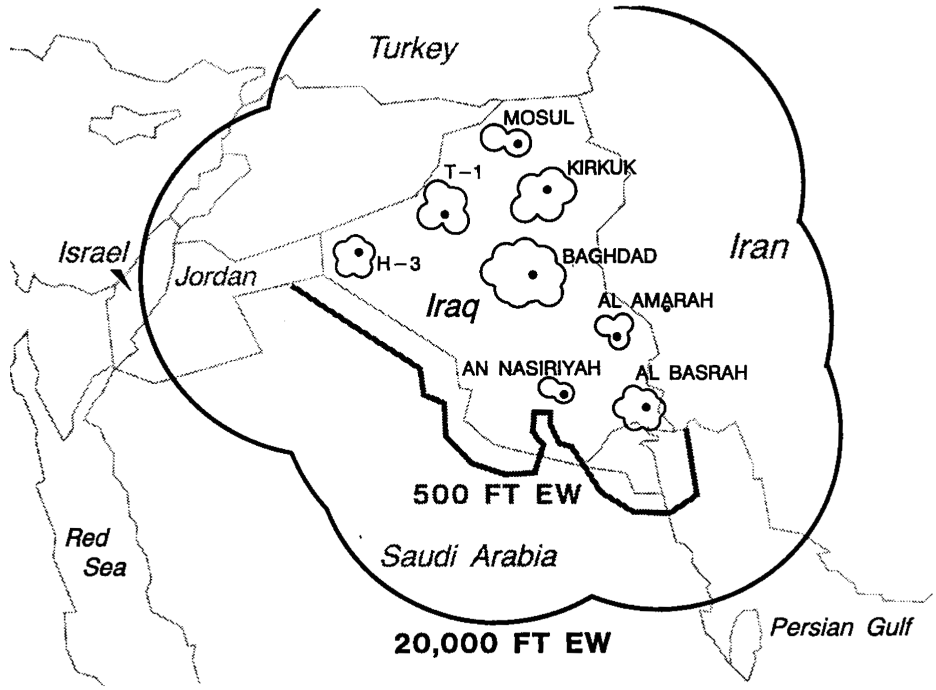


Figure 34: SAM and Early Warning (EW) Coverage of Iraq in 1991 [105].

Additionally, over 7,500 pieces of AAA including the 23-mm ZSU-23/4 and the 57-mm ZSU 57/2 self-propelled systems protected high value targets [447]. As demonstrated on several notable occasions during the Gulf War, the threat environment below 10,000 feet creates an envelope where it is impossible to safely operate. A lower-bound for operational altitude is defined at this level. While operation at medium and high altitudes contributes to survivability, these heights limit accuracy, target identification, and target acquisition. "Medium and high-altitude tactics also increased the exposure of aircraft sensors to man-made and natural impediments to visibility" such as smoke, camouflage, foliage, sandstorms, fog, haze, clouds, and high humidity [163].

While the properties of each type of SAM can be defined using References [447] and [327], to avoid sensitivity concerns with identification of realistic system properties, SAM sites are given uniform properties representative of 1991 SAM systems. Because, the maximum range of surface to air missiles and their target tracking radar systems are parametric variables in the simulation, current and future threats can be analyzed as well. The process of calibrating the simulation for the 1991-level IADS is discussed in Appendix E.2.

The Iraqi air force contained more than 700 fighters, including less than 350 third-generation (MiG-23, MiG-25, and Mirage F-1) or fourth-generation (MiG-29 and Su-24) fighters [457]. The remainder of the aircraft were older Soviet/Chinese technology, operated by relatively unskilled pilots. Hussein's conservative doctrine preferred to keep his air forces as a reserve for the final defense of Baghdad. In the Iran-Iraq war, Iraqi tactics were dominated by survival, even when the numerical advantage was on their side. Despite intensive use of jamming and countermeasures by U.S. forces, during the war "the Iraqis employed few, if any, electronic countermeasures and presented almost no air-to-air opposition" [164]. Based on the performance of U.S. assets in air-to-air engagements over the past fifteen years, it is reasonable to assume that the primary threat to U.S. air assets comes not from hostile fighters but from ground-based SAMs and AAA. According to Khalilzad, "loss rates have been reduced by more than two orders of magnitude since World War II," with the loss rate during Operation Desert Storm and Operation Allied force at 0.4 and 0.1 aircraft lost per 1,000 sorties respectively [236]. As Bolkcom notes, "Since Operation Desert Storm, 100% of all U.S. combat aircraft losses have been due to enemy air defenses. No U.S. aircraft has been lost to an enemy aircraft since 1991" [66]. Therefore, to reduce the simulation burden, offensive counter air (OCA) and defensive counter air (DCA) sorties are not simulated.

Theater Ballistic Missiles (TBMs), "with a mix of warheads have the capability to alter the course of campaigns. Thus, Joint counterair operations are a key to successful future campaigns" [490]. Due to their potential to distribute NBC weapons, ballistic missiles are a major target of interest in the LRS scenario. While fixed missile launch sites and production facilities can be often be detected using overhead imagery, time critical targets such as mobile TBM launchers also pose a great threat to friendly forces. Approximately 600 Scud-B missiles and twenty-two mobile transporter-erector-launchers (TELs) were purchased from the Soviet Union between 1976 and 1979 [104]. An example of a mobile TEL is shown in Figure 35.

Figure 35: Scud Mobile Transporter-Erector-Launcher (TEL) [462].

Approximately 200 of these missiles were fired in the Iraq-Iran War of the 1980's. The reported number of TELs, 37, includes an estimated 15 indigenously constructed launchers. During Persian Gulf conflict, Iraq fired "88 modified Scuds, 42 toward Israel and 46 at Saudi Arabia and other Persian Gulf states" [457]. "The Iraqi version of the Scud, the Al-Hussein, had a circular error probability of more than 2,000 meters and carried less than 180 kilograms of high explosives" [105]. Despite General Schwarzkopf's assertions that the Scud missiles posed no military significance, on February 25, 1991, an Iraqi Scud missile hit an Army barracks in Dhahran, Saudi Arabia, killing 28 Americans and wounding nearly a hundred causing the largest single loss of life in the conflict [85]. Although post-war accounts noted that the U.S. counter-Scud effort "did in fact reduce Iraq's ability to launch missiles," historical records indicate that no mobile Scud launchers were confirmed destroyed in Operation *Desert Storm* [345].

In addition to the number of TELs destroyed in counter-scud operations, one measure of effectiveness of critical interest in the simulation activity is the number of Scud missiles fired. Technologies which either suppress Scud activity or eliminate launchers, missiles, fuel depots, and other processing facilities support the GSTF mission of eliminating potential WMD delivery systems.

5.2.1.6 Own Courses of Action: Defining Specific Analysis Scenarios

> *"The first thing for a commander in chief to determine is what he is going to do, to see if he has the means to overcome the obstacles which the enemy can oppose to him, and, when he has decided, to do all he can to surmount them."*
>
> -Napoleon, Maxim LXXIX

The Operation *Desert Storm* scenario was chosen to demonstrate the proposed methodology primarily due to the availability of data to assist in the validation phase. As previously mentioned, the development a robust portfolio implies that a technology suite is effective across a number of scenarios. Therefore, in addition to simulating the relevant events of the opening phase of the Gulf War, the same backdrop can be used to demonstrate multiple scenarios and assess the robustness of a technology portfolio. Three scenarios that test different aspects of a robust LRS capability are a decapitation strike, an attack against hardened deeply buried targets, and time critical target strike.

Decapitation Strike

Originally defined in the context of nuclear war, a decapitation strike is a first strike attack whose operational aim is to neutralize the command and control structure of an adversary in an attempt to severely degrade an attempt at response [430]. This strategy was used on March 20, 2003 when the United States launched an attack on leadership targets in Iraq in an attempt to "undermine Saddam Hussein's ability to wage war" [81]. In addition to more than 40 Tomahawk missiles launched from three U.S. Navy vessels, "two U.S. F-117A Stealth fighters were also involved in the attack, dropping 'bunker buster' bombs on targets" [238]. This marked the first combat use of the EGBU-27, a laser-guided bomb capable of support from both inertial navigation and GPS [203]. The choice of this munition was critical, as Baghdad was obscured by low-level clouds on the morning of the strike. The entire mission was planned and executed in only five hours, which Pilot Mark J. Hoehn noted was "unprecedented." Hoehn's F-117, landing at al Udeid airbase in Qatar is shown in Figure 36.

In another "pop-up" opportunity to strike at Saddam Hussein on April 7, 2003, a B-1B bomber loitering in Western Iraq hit a target in downtown Baghdad only 47 minutes after being assigned. According to officials, "most of the 47 minutes was consumed in the discussion about whether or not to attack" [203]. The B-1B flew to the target at subsonic speeds in only twelve minutes and delivered hard-target penetrating BLU-109B/GBU-31 JDAMs shown in Figure 36.

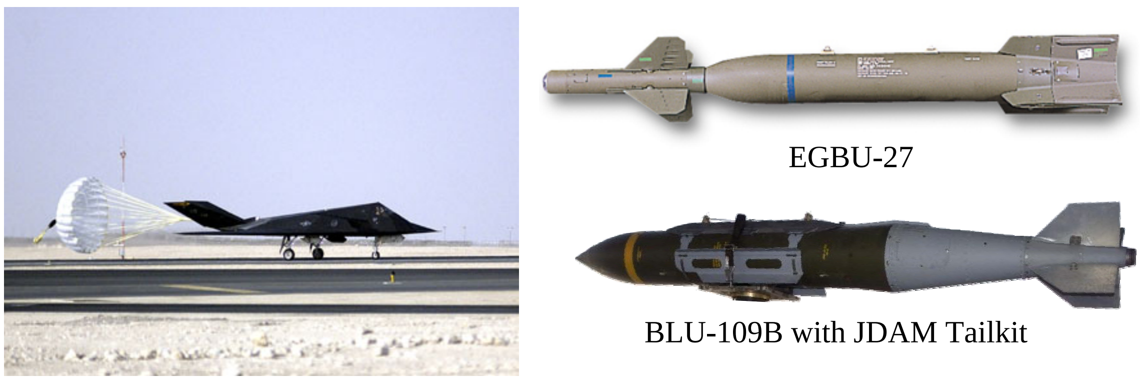


Figure 36: F-117A *Nighthawk* Returning from a Decapitation Strike (left) and Munitions Used in Two Decapitation Strikes (right) [203, 439].

According to national security analyst James S. Robbins, "this technique is only effective against dictatorships, in which a single person or small group comprise the center of gravity" [352]. A decapitation strike can be simulated by prioritizing leadership and C3 targets while essentially ignoring air defenses and other strategic targets. The time critical nature of the decapitation strike can be measured by simulating only a very short simulation time (on the order of one hour) to assess whether or not the leadership targets were successfully struck before relocating to more secure facilities.

Hardened Deeply Buried Target Strike

One of the primary missions for the Global Strike Task Force is the defeat of hardened deeply buried targets (HDBT) [326]. According to the Department of Defense, future "adversaries will employ sophisticated deception tactics, including hard and deeply buried targets, which pose significant identification and force application challenges for our forces" [453].

The National Research Council, defines hard and deeply buried targets as "all types of intentionally hardened targets, either aboveground or belowground, that are designed to withstand or minimize the effects of kinetic weapons." The Defense Intelligence Agency (DIA) estimates that there are "10,000 known or suspected hard and deeply buried targets worldwide... About 20 percent have a major strategic function, and of those, about half are in or near urban areas" [309]. According to the Air Force Scientific Advisory Board, "in general, as the military importance of a target increases, the number of targets decreases, and the header and deeper they will be buried" [160]. While most of these targets are between 100 and 400 meters deep, several are up to 700 meters deep and surrounded by granite or limestone. Examples of several types of hardened deeply buried targets are shown in Figure 37.

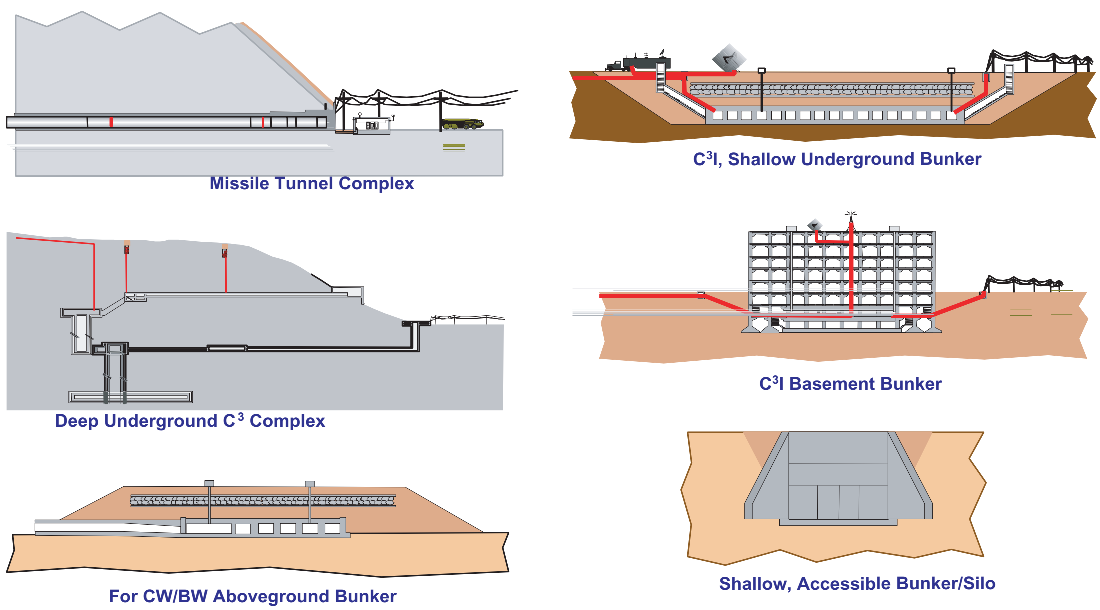


Figure 37: Examples of Strategic Hard and Deeply Buried Targets [309].

The DoD states that current "weapons development efforts prove inadequate to overcome adversary use of hardened and deeply buried facilities to protect key capabilities" [455]. For example, while Tomahawk missiles were used in the opening days of Operation Desert Storm, the Gulf War Air Power Survey notes that "the small size of the missile warhead as well as its inability to penetrate hardened targets had limited its effectiveness" [105].

Since the end of the Persian Gulf War, efforts have been made to produce weapons systems to overcome this shortfall. In May of 1997, the GBU-37B, a 4,700 lb GPS-aided "bunker buster" was test dropped from a B-2A [334]. In addition to limited operational use in Kosovo, in 2003, two GBU-37B munitions were dropped on an Iraqi communications tower during Operation *Iraqi Freedom* [254, 108]. Advanced penetrator concepts such as hypersonic weapons, nuclear Earth penetrators, and large conventional munitions are also being considered for conventional Prompt Global Strike Capability [160, 309, 384, 59]. According to Air Force historian Dr. Richard Hallion, future concepts may also include "robotic micro-munitions to attack deeply buried hard targets" [189].

A HDBT mission can be simulated using a limited strike against one or more super-hardened targets. An example of one such target is the National Air Defense Operations Center (ADOC) in downtown Baghdad [139]. According to the DoD, "the ADOC maintains the overall air picture and establishes priorities for air defense engagements" [458]. It also connects to the five Sector Operations Centers (SOCs) that control air defense operation and BM/C4ISR assets. ABC military analyst Anthony Cordesman also notes that "the SOCs cannot communicate effectively once the ADOC [is] destroyed or deactivated" [107]. A functional or physical kill of this target would allow friendly assets to overwhelm the defenses of each sector in isolation.

In contrast to the decapitation strike scenario, the attribute of timeliness is outweighed by the need for a large payload capacity and survivability in the presence of anti-access threats.

Time Critical Strike

According to Brown, one of the critical areas for technology evaluation for LRS is the ability to respond to time critical threats [76]. The Air Land Sea Application (ALSA) Center defines a time critical target (TCT) as "a lucrative, fleeting, land, or sea target of such high priority to friendly forces that the JFC or component commander designates it as requiring immediate response" [427]. Such targets may also be called flex targets, emerging targets, fleeting targets, mobile targets, or time-sensitive targets [272]. Notable examples include C2 vehicles, mobile terrorists, SAMs, mortar teams, and Scud missile TELs.

The primary technical challenge associated with the time critical strike mission is the extreme importance of timeliness and compression of the kill chain. In 1991, U.S. Air Force assets were unable to destroy mobile Scud launchers that could "shoot-and-scoot" in less than 90 minutes [355]. Commander of the Air Force Materiel Command, General Bruce Carlson noted that although progress was made during the 1990's, "TCTs remained a considerable challenge" during Operation *Allied Force* in Kosovo [86]. According to General James Morehouse, the "key to achieving the right effects is [the] warfighter having the right information at the right time to make the right decisions. Seamless, integrated C4ISR is [an] enabler for making that decision" [301]. The time critical targeting problem is also confounded by weather, communications delays, and administrative tasks [255, 204]. Detection and tracking of present day TCTs is further degraded when time critical targets also use Camouflage, Concealment, and Deception (CC&D) [226].

Despite the notable capability gap with respect to TCTs, in 2001 General Jumper stated a desire to reduce the characteristic time of the kill chain to "single-digit minutes" [232, 204]. A thirty minute timeline used for the 2002 Joint Expeditionary Force Experiment (JEFX 02) is shown in Figure 38.

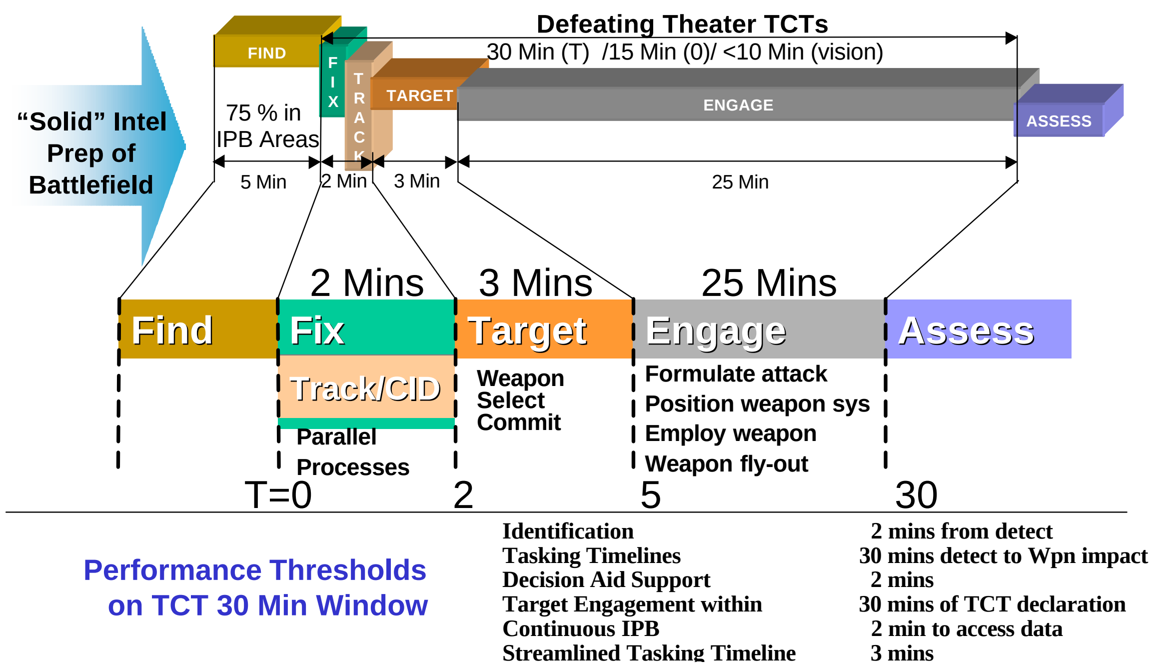


Figure 38: Example Timeline for TCT Attack [301].

The location of targets depends on "solid" intelligence preparation of the battlefield. Based on this figure, 83% of the kill chain time is allocated to the engage function, which details positioning and employment of the weapon; however, the decision time allocated to the fix, track, and target functions and the "seams" between stages is also extremely compressed [204]. Technologies which contribute to better target location information reduce the characteristic time of the find function, while integrated C4ISR technologies compress the time allocated to the fix, track, and target functions. Different schools of thought exist on how to reduce the time allocated to the engage function, with CONOPS that vary between high-speed aircraft that minimize flight time or "area dominance" concepts where a persistent, survivable platform loiters within denied airspace and waits for instructions.

To assess the effectiveness of an LRS architecture at the TCT attack mission, elements of these disparate architectures can be compared. Time critical targets will consist of mobile Scud TELs (see Figure 35) capable of moving, firing, and hiding according to the activity diagram shown in Figure 67. Section 5.5.9 describes model development to support the TCT attack mission, which differs dramatically in character from the other two scenarios. In addition to MoEs related to targets killed and time to target, MoEs for this scenario also include enemy TBMs fired.

GSTF Attack Three Day Scenario

Each of the three aforementioned scenarios are designed to test different aspects of an LRS architecture, as well as subsystem, system, and architecture technologies that enable LRS capabilities across the characteristics identified in Section 2.4. In addition to these three defined scenarios, a fourth scenario was added to test the integration of all LRS elements into a single long-duration campaign. Unfortunately, the complexity and number of elements in the overarching scenario including Tomahawk cruise missiles, ALCM's, area dominance munitions, and LRS assets confounded the results. For this reason, the "all-out-war" scenario was reduced in complexity to represent six LRS assets operating against 420 targets over a period of three days. This scenario is called the "Global Strike Task Force (GSTF) Attack Three Day Scenario" and is intended to analyze the impact of technologies with respect to multiple sorties over an extended period of time. A summary of the major challenges of each of the four scenarios is shown in Figure 39.

			Decapitation	HDBT	TCT Attack	*GSTF 3 Day*
LRS	Characteristics	Long Range	✓	✓		✓
		Persistence			✓	
		Rapid Responsiveness	✓		✓	✓
		Flexible Payload	✓	✓	✓	✓
		High Survivability	✓	✓	✓	✓
		Situational Awareness			✓	
Other	Attributes	Revolutionary Architectures			✓	
		Target Hardness		✓		
		Mobile Target			✓	
		Fleeting Target	✓		✓	
		Multiple Sorties				✓
		Very Large Payloads		✓		

Figure 39: Summary of Characteristics for Four LRS Scenarios.

5.2.1.7 Analysis of Courses of Action and Decision Making

A unique feature of the methodology proposed in this dissertation is the use of machine learning and surrogate models to address the final three branches of the "estimate of the situation" depicted in Figure 25.

Since World War II, the offensive power brought to bear by U.S. forces has limited the enemy's courses of action to primarily defensive maneuvers with the notable exception of TBM launches and suicide attacks. For this reason, cognition models have been developed to support the actions taken by the IADS, prioritizing threatening assets and coordinating the attack against them. TBM launchers have been pre-programmed to move, hide, and fire at specified intervals. A major coalition course of action is to eliminate these launchers before they fire and to destroy infrastructure across all target sets.

The decisions on the courses of action for targeting and weaponeering are primarily performed by the intelligent battle manager, whose development is summarized in Section 5.5.1. The specific elements of the "decision" branch in Figure 25 are defined below:

- **Who:** The countries involved in the conflict are defined above in Section 5.2.1.4. In the simulation, the actual actors and their actions are defined using intelligent agents that mimic human-like decisions.

- **What:** Coalition aircraft are tasked to engage targets based on the capabilities they provide. Decisions on what to attack are made by the Meta-General. The apportionment of assets will be guided by the air order of battle defined in Figure 2 with the addition of LRS assets as an element under test.

- **When:** The order in which to attack target sets is defined using the QFD method of target prioritization summarized in Section 5.4.1.

- **Where:** While the location of hostile assets has been defined as described in Section 5.2.1.3, friendly assets are positioned at Khamis Mushait airbase in Southern Saudi Arabia, the strike base for the F-117A as defined by the Conduct of the Persian Gulf War report. Most of the support forces are not simulated due to the weak coupling between the LRS and these simulation elements.

- **How:** Cognition models guide intelligent agents in the performance of a mission according to realistic constraints on their operation, noting that the tactics, techniques, and procedures manuals that would define the exact operational limits more definitely are not publicly available. Cognition models used in the simulation activity are detailed in Section 5.3.2.1.

- **Why:** The philosophical reasons for the conflict are defined by policymakers at the highest levels of government. The main reason for coalition involvement in the 1991 Gulf War was to eject Iraqi forces from Kuwait, restore its legitimate government, and eliminate Saddam Hussein's ability to threaten his neighbors. The strategic objectives that motivate involvement in the conflict are summarized in Section 5.4.

Using the "estimate of the situation" to set the stage and establish the boundaries of the scenario to be used for the proof-of-concept activity, a simulation was created that uses this scenario as the "game board" for the quantitative evaluation of technologies and systems. The details of the models developed to support this simulation are described in subsequent sections.

5.2.2 Establish Measures of Effectiveness (MoEs)

A Measure of Effectiveness (MoE) "provides a standard by which it can be establish how well some thing achieves the purpose for which it is intended" and is the primary output of the simulation and analysis activity [381]. They provide a measurable way of evaluating how well a proposed DOTMLPF[10] solution provides capabilities to achieve a desired result. There are numerous synergistic definitions including:

- A measure provides the basis for describing varying levels of task performance [428].
- "standards against which the capability of a solution to meet the needs of a problem may be judged" [382]
- MoEs are "the metrics by which a customer will measure satisfaction with products produced by a technical effort" [216].
- The Department of Defense further clarifies MoEs as "a qualitative or quantitative measure of a system's performance or a characteristic that indicates the degree to which it performs the task or meets a requirement under specified conditions" [370].

These metrics should be the top level goals to which the system-of-systems is designed. To quantify the impact of solutions, MoEs should be *measurable quantities* with *defined units* where appropriate. Furthermore, the MoEs must be *calculable* using the simulation tool. While Roche and Watts note that formulating "good" MoEs is a difficult task that has continually confounded operations researchers [353], some guidelines for formulating appropriate MoEs are given in Reference [428]. The definition of MoEs is critical because

[10] In the JCIDS parlance, DOTMLPF stands for "Doctrine, Organization, Training, Materiel, Leadership, Personnel, and Facilities," all possible solution domains to close a capability gap. This work primarily focuses on materiel solutions in the system and technology domain although tactics may be classified under doctrine or training.

they provide an objective means to measure a capability. On the other hand, Measures of Performance (MoPs) may be tied to a specific physical implementation, making it difficult to compare dissimilar concepts. According to the Defense Acquisition University, MoPs "quantify a technical or performance requirement directly derived from MOEs" [130]. When the simulation environment is constructed correctly, different MoPs may be used to calculate the same MoEs. Consistency in the transfer function between MoPs and MoEs enables evaluation of dissimilar concepts with respect to the same top-level capability metrics.

The relationship between MoEs, MoPs, and lower level Technical Performance Parameters (TPPs) as defined by the Defense Acquisition University is shown in Figure 40. Systems engineering decomposition techniques define the relevant MoPs for one or more MoEs. A similar decomposition can be performed to identify the relevant TPPs. Modeling and simulation is used in this research to quantify the impact of changes in TPPs on top-level capability metrics.

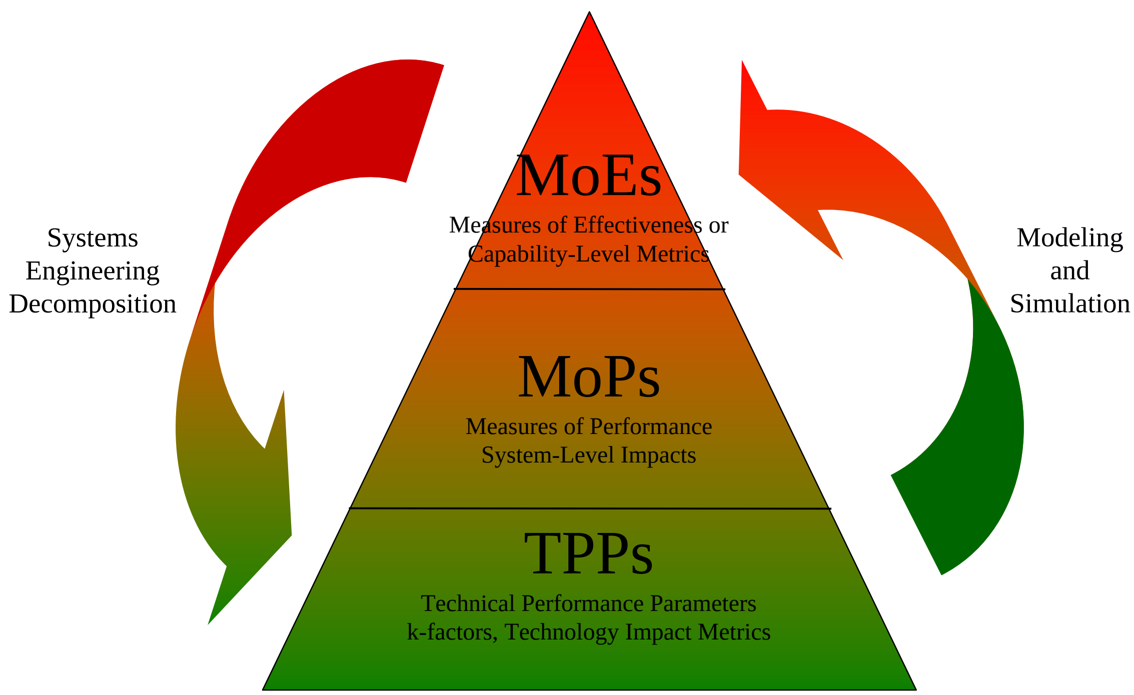


Figure 40: The Relationship Between Effectiveness, Performance, and Technologies.

Since the effectiveness of a proposed DOTMLPF solution relies on the numerical value of MoEs, the identification of the appropriate measures of effectiveness to evaluate capability gaps is critical. Matsumura notes that it is “hard to get beyond the measures of success used in the Cold War” where the primary objective was aimed at halting a massive armor invasion of eastern Europe [276]. Although the term was first used by Morse and Kimball in the 1940’s, Green and Johnson claim that an explicit theory for the development of MoEs does not exist [303, 184].

MoEs can be identified through references to Air Force doctrine, determined from brainstorming desired outcomes (for a given capability or capabilities), or by examining the simulation tool to find information that is available and then determining whether that information is of interest. A list of MoEs for the Long Range Strike testbed activity were determined using all three methods and are enumerated as follows:

- **Duration of Conflict:** The total time in hours that elapses from the assignment of GSTF assets to the conclusion of strategic operations. Conclusion is defined as the point at which attrition levels across all target sets have reached a user-defined percentage[11].
- **Number of Strikes Flown:** A sortie is defined as “an operational flight by one aircraft” [468]. A strike is “an attack which is intended to inflict damage on, seize, or destroy an objective” [468]. As noted in the Gulf War Air Power Survey, the distinction between strikes and sorties is confusing [106]. For the calculation of this measure of effectiveness, a strike is defined as “each time an aircraft successfully releases a weapon toward a unique target.”
- **Munitions Fired:** If each aircraft delivers only a single munition to each target, the number of munitions fired is numerically equal to the number of strikes flown. This simplifying assumption is used for the proof-of-concept demonstration.
- **Cost of Munitions Fired:** Every munition fired has a cost associated with its replenishment. Reference [106] provides an excellent reference for weapon prices in the 1991 time frame. The cost of each weapon fired is recorded.

[11] For a discussion of military target sets, please see Section 5.2.1.3.

- **Number of Red TBMs Fired:** Since Theater Ballistics Missiles (TBMs) are preprogrammed to fire at allies in predefined intervals, minimizing the number of TBMs fired is an MoE that represents effectiveness at the TCT attack mission.
- **Number of Blue Units Lost:** The number of blue assets of all types lost in a user-defined time interval.
- **Number of Blue Aircraft Lost:** The number of blue aircraft platforms of all types lost in a user-defined time interval.
- **Number of Blue LRS Aircraft Lost:** The number of blue Long Range Strike aircraft lost in a user-defined time interval.
- **Percentage of Red TCTs Killed:** In addition to attrition numbers related to TBM launchers and production, the number of mobile TBMs in the simulation is known *a priori*[12]. The percentage of these time-critical targets killed at a user-defined time interval is tracked to assess effectiveness of the LRS architecture at the time critical strike mission.

[12] Even though this may not be true in reality.

5.3 Step 3: Problem Definition and Creation of a Simulation Environment

The creation of a modeling and simulation environment with the necessary systems, linkages, physics, cognitive models, and appropriate level of fidelity at each hierarchical level is arguably the most time consuming step in the process shown in Figure 23.

A methodology for making decisions at the system-of-systems level first requires an analysis environment that allows a holistic view of the entire system-of-systems so that design studies can be performed on various system architectures and the elements of those architectures. As noted by Soban, a conceptual model or "plan of attack" aids a designer in identifying which models need to be designed to accurately represent the system-of-systems [378]. A paradigm for the modeling and simulation of complex systems containing the necessary elements in the system-of-systems formulation is shown in Figure 41.

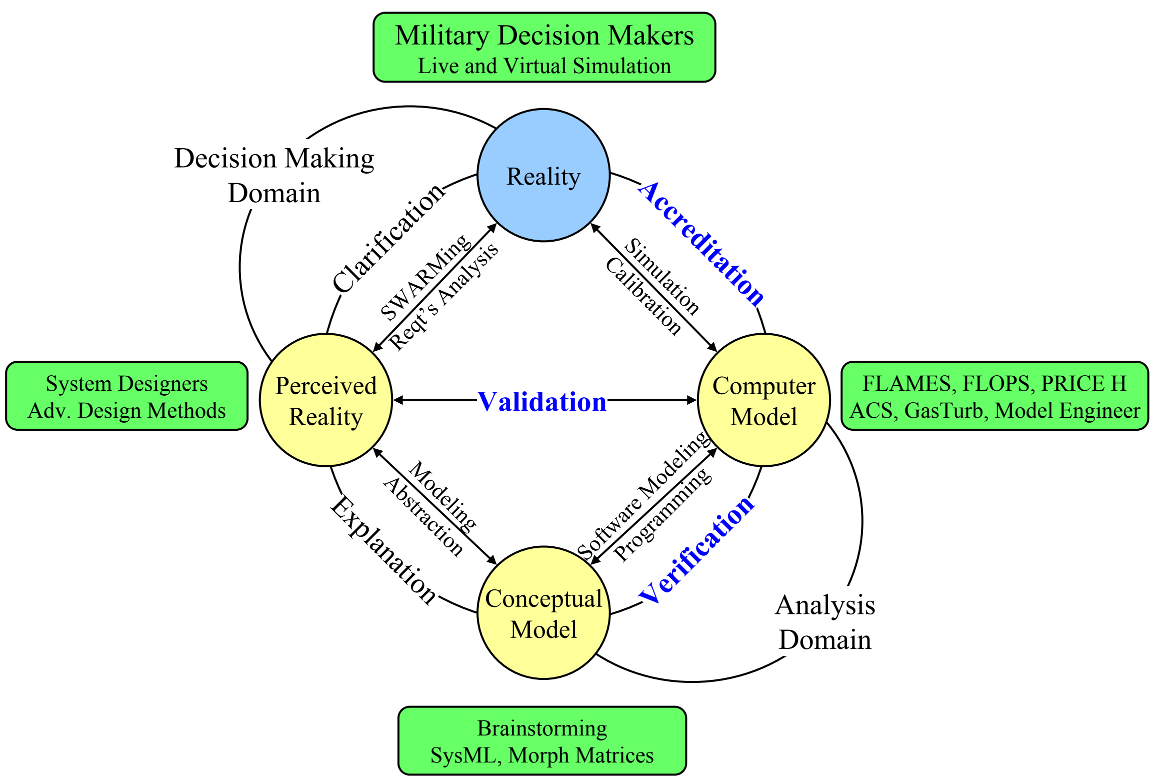


Figure 41: Modeling and Simulation Paradigm (modified from Reference [258]).

A simulation is based on reality. Due to the nature of this work, not all "real" information is available for analysis (eg: classified, FOUO, etc.) As a result, through a process called *clarification*, a perceived version of reality is constructed. This perceived vision of reality can also include the scope of a reduced order problem that represents a real world phenomenon. Literature searches and the SWARMing technique facilitate progress in the clarification phase.

Next, through *explanation*, a conceptual model is created. A conceptual model is an abstraction of an experiment to be performed that defines the process to be followed, necessary elements of the experiment, important inputs, and expected outcomes of the experiment. This process follows the steps of the scientific method. This prevents the creation of unneeded models and is analogous to modeling in the software community in which modules are laid out in a pseudocode form prior to actual programming [155]. Storyboards used by animators in the motion picture industry are another analogy to the conceptual model [83]. Creating a detailed conceptual model avoids the development of overly specific mathematical models that have hard-wired assumptions or insufficient degrees of freedom to be useful for real problems. Ideally, the conceptual model is independent of the simulation framework chosen and physics-models used: it identifies what is needed to perform the simulation under defined standards to accomplish an objective. Brainstorming tools, morphological matrices, the Systems Modeling Language and the systems engineering methods facilitate the creation of a conceptual model (see Section C.6).

Furthermore, Zaerpoor and Weber add: "In making the conceptual model, one starts with the questions of interest and conceives of the leanest physical experiment that would result in satisfactory answers for the questions" [494]. This observation is critical to the design of systems-of-systems. Without somehow narrowing the space of interest, the design problem is intractable. On the other hand, Ilachinski notes that decomposition of complex systems can sometimes destroy the emergent behavior derived from the complicated interactions at the lowest hierarchical level [213]. A balance between these two schools of thought is needed to include high-fidelity models where appropriate and lower fidelity effects-based models when the impact on MoE variability is low. The challenge of balancing fidelity and scope is one of the fundamental problems of system-of-systems design.

The next three steps, verification, validation, and accreditation (VV&A) process are defined by DoD directive 5000.61 and must be performed for any simulation tool certified to operate within the acquisition community [419]. Since the environment constructed to validate the proposed methodology will not be directly used for acquisition decisions, it does not require the costly and time consuming VV&A process; however, efforts must be undertaken to ensure that the models are correct and that they are being used correctly.

The *verification* process is one by which a computer model is made to match the conceptual model, and essentially answers the question, "are the models correct?" This code-intensive phase uses the FLAMES framework and will require extensive testing and evaluation to ensure that the computer models are doing what they are supposed to do. The computer model is an instantiation of the conceptual model and "allow[s] for the changing of the values of a parameter or set of parameters and then determining the effect that this has on the variables of interest" [41].

When the computer models match perceived reality, this is called *validation*. Essentially, the validation phase answers the question "are the models being used correctly?" If the validation phase is successful, then the simulation is an accurate representation of the real-world from the perspective of the intended uses of the model or simulation. If the simulation does not match perceived reality, the lower half of the process shown in Figure 41 must be repeated.

The process by which a model is certified (matches reality) is called *accreditation*. This final step is the most time consuming and expensive phase and results in a model that is certified for acquisition purposes. The accreditation phase is not needed for the proof-of-concept exercise.

In practice, the VV&A process is difficult for agent-based constructive simulations of future concepts for which there is no empirical database to validate against. This situation is typical for advanced development labs and the best practice for VV&A for this type of simulation is to independently assess each of the physics-based models and ensure that the physics is being modeled correctly. For cognition models of agent behavior, the agents must be tested under various conditions to ensure that they follow the correct decision paths in isolated tests. It is then usually inferred that the aggregated behavior is as correct as possible. Periodic evaluation of results is helpful to determine whether this assumption is correct.

Within step three, there are a number of subfunctions that must be performed to assemble a validated simulation environment that can successfully demonstrate the proposed methodology as applied to the sample problem. These steps are (1) creating the conceptual model, (2) creating the physics-based models defined by the conceptual model, and (3) assembling a simulation and verifying that the modeling and simulation environment can be applied to the example problem.

5.3.1 Step 3.1: Creating the Conceptual Model

The first step in creating the conceptual model is identifying the elements that must be in the simulation environment to model the Long Range Strike architecture. The second step is the creation of models in support of this conceptual model, and the final step is the development of a simulation that uses these models with the scenario(s) defined in Section 5.2 to validate the proposed methodology.

5.3.1.1 A Process for Architecture Definition

The first step in creating a modeling and simulation environment is to identify the characteristics of future LRS architectures. An architecture is described by both systems and the means to link them together: both physical and functional attributes. Defining the modeling and simulation environment begins with defining these architecture elements.

Creating a system architecture from a top-down, capability focus can be difficult. Engineers have a tendency to jump immediately to physical systems because these “widgets” are what designers are most familiar with. Since the goal of a systems engineering process is to define the attributes of this equipment that maximizes value to the customer, a top-down decomposition process should begin using the top-level goal of “secure and defend the nation,” which is the “first and fundamental commitment of the Federal Government” [80]. This top-down decomposition is notionally depicted in Figure 42. The Quality Function

Deployment technique forms the backbone of this decomposition of requirements to systems [135, 325].

Since is desirable to identify capabilities in relation to specific challenges to national security, the first QFD matrix in Figure 42 relates a single goal (Provide National Security) to a number of national security challenges. The Air Force refers to these as Focused Long Term Challenges (FLTCs). The roof of QFD #1 shows how different challenges may be positively or negatively correlated. The second QFD matrix identifies how joint capabilities can answer the identified national security challenges. The roof of this QFD can identify redundant or complimentary capabilities, and the joint focus of this matrix allows the formulation of joint strategy and doctrine to maximize effectiveness across military services. This is consistent with the focus of the JCIDS process. The superset of all military capabilities is defined by the Universal Joint Task List (UJTL) [460]. Air Force tasks are defined by the Air Force Task List (AFTL) [428] and Air Force doctrine documents [431, 433]. While these capabilities encompass the range of air warfare, for this study, only Long Range Strike capability, a subset of the AFTL also identified by the AFRL Vehicles Directorate (Section 1.3), is examined in detail.

Next, a third QFD matrix can be formulated that relates joint capabilities to operational activities. For this study, the operational activities are defined by the generic elements of the Air Force kill chain, as summarized in Section 5.3.1.3. The fourth QFD shown in Figure 42 shows how the elements of the kill chain can be performed by various system functions. This QFD is identical in purpose to the DoDAF SV-5 operational activity to system function traceability matrix. While these system functions are *independent of physical systems*, it is often useful to look to the "as-is" military architecture to determine generic functions that are performed by systems today. For example, the Air Force Fact Sheet for the E-3 Sentry aircraft says, "The radar combined with an identification friend or foe subsystem can look down to detect, identify and track enemy and friendly low-flying aircraft" [440]. From this statement, the system functions *detect targets*, *identify targets*, and *track targets* can be defined. To interoperate with existing military assets, one or more elements of the Long Range Strike system architecture must perform these functions or their equivalent.

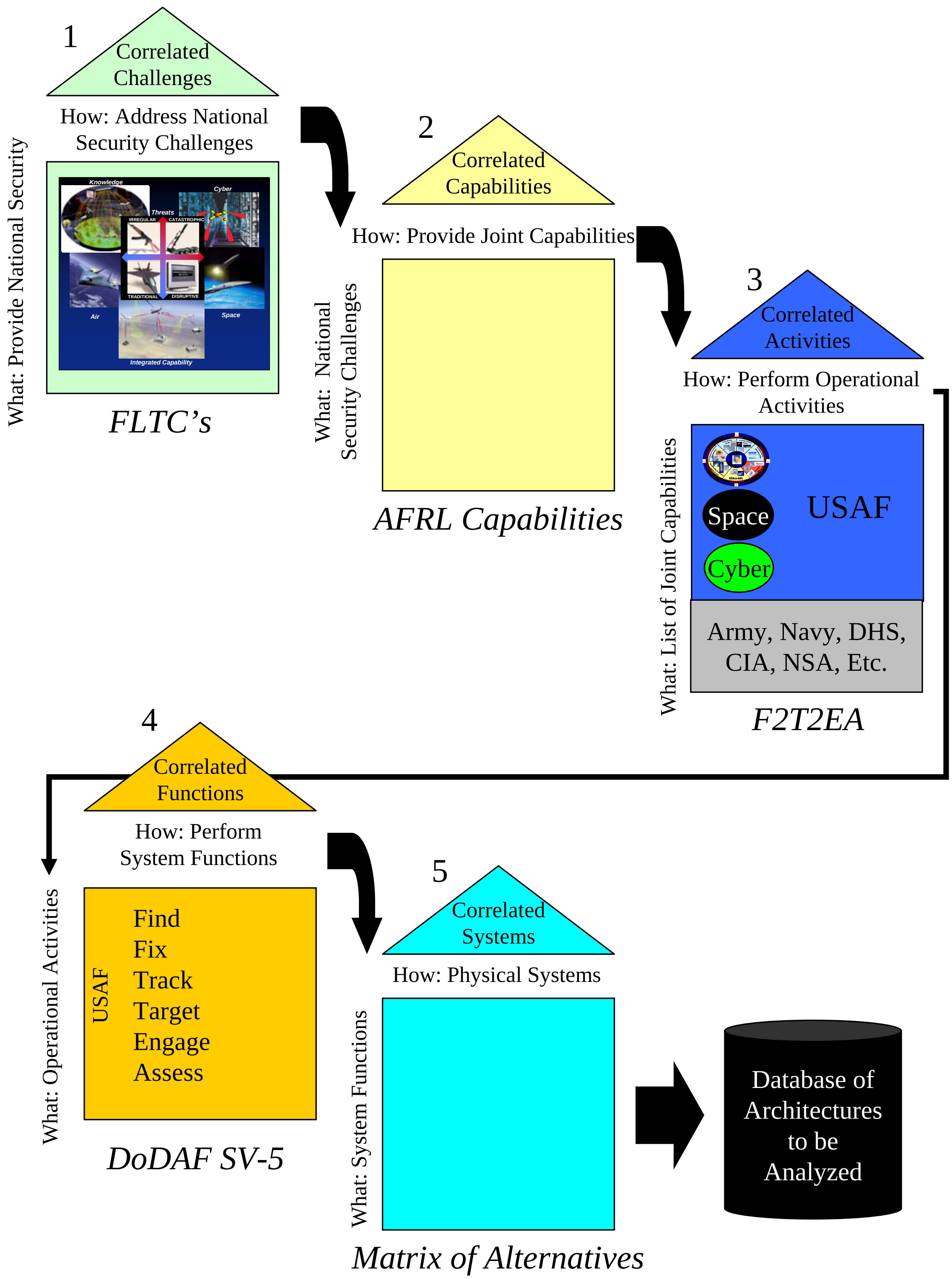


Figure 42: Quality Function Deployment for a Long Range Strike System Architecture.

Furthermore, the establishment of these system functions aids in the creation of cognitive models that allow assets to perform these functions. The fifth QFD matrix is actually a matrix of alternatives (see Section C.5.1). Instead of decomposing system elements against system attributes, the rows of the matrix of alternatives are the system functions that must be performed and the columns indicate different physical systems that can perform these functions. Selecting one or more elements in each row defines a single architecture. Architectures of interest are then analyzed using a physics-based modeling and simulation environment.

It is evident that, given enough resources, all possible architectures could be examined over all functions, all operational activities, across all domains, capabilities, branches of service and against all strategic challenges for every potential threat. The design space for this "ultimate architecture" that satisfies all customer requirements for all anticipated threats is beyond the computational power of any known organization. This work holds the elements of QFD matrices 1-3 constant and focus on identifying system functions and physical systems that can perform these functions. Intelligent agents are used to examine the infusion of technologies to a defined Long Range Strike architecture; however, the environment assembled for this study is available for future work to expand these additional degrees of freedom.

5.3.1.2 Using the DoDAF OV-1 to Describe the Architecture

Given the desire to examine Long Range Strike capability, it is first necessary to lay out the "big picture." The DoDAF OV-1 High-Level Operational Concept Graphic [137], shown in Figure 43, pictorially describes what the LRS architecture is intended to do. Software such as MagicDraw UML, Rational® Software Modeler, or ARTiSAN Studio can be used to define an architecture consistent with the OV-1 while also laying the groundwork for implementation of FLAMES code. The OV-1 shown in Figure 43 defines the architecture in general terms and was created through a literature search of public documents and a brainstorming activity described in the subsequent section.

The purpose of the architecture is to strike ground targets over long ranges. The primary

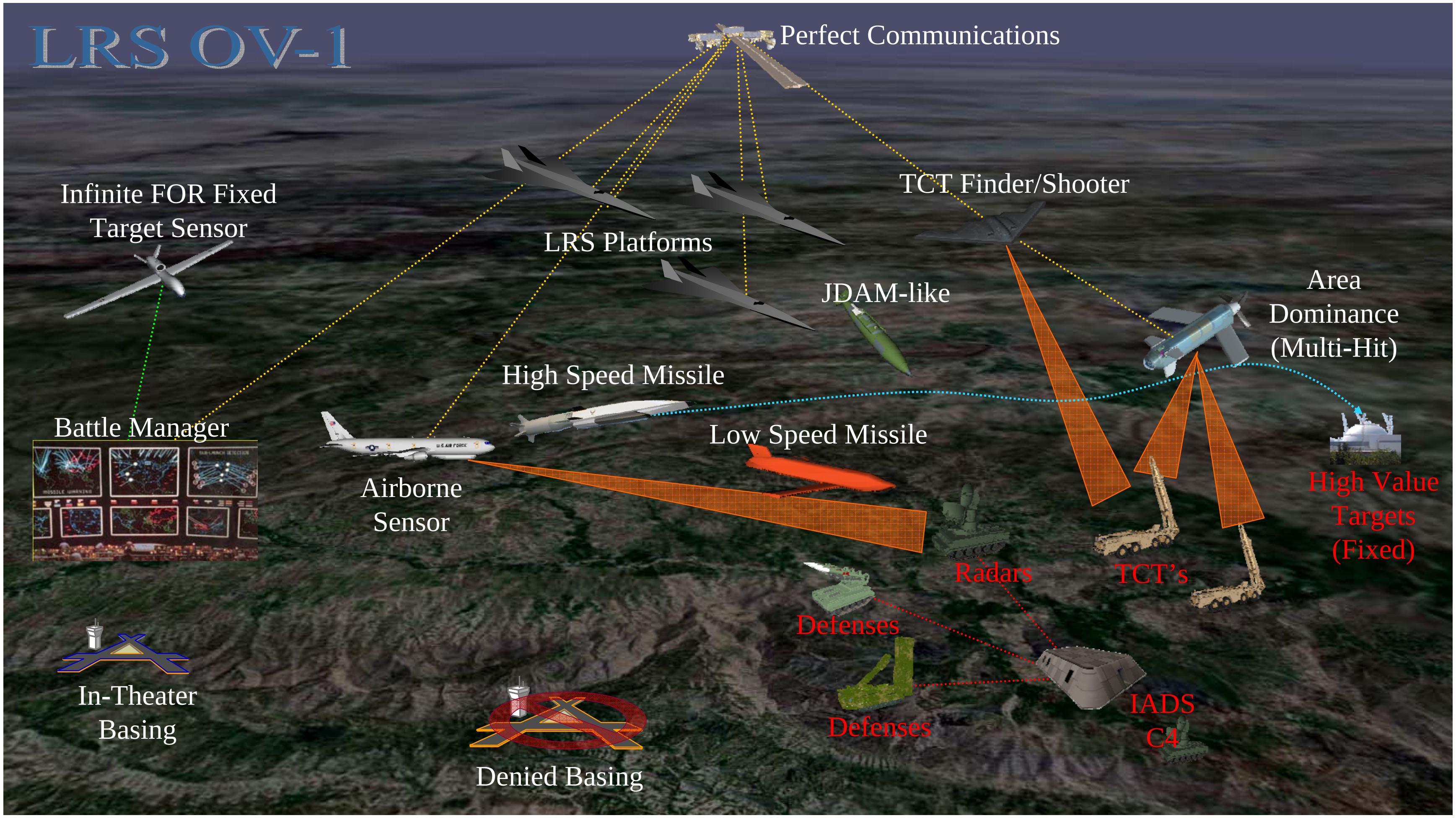


Figure 43: Notional LRS OV-1 Operational Concept Graphic.

"actors" in the simulation are the LRS platforms that are the primary element under test. These platforms can fire different types of munitions including JDAM-like guided bombs, low speed cruise missiles, and high speed supersonic and hypersonic missiles. Additional shooters include a B-2 like TCT finder/shooter. This asset has an onboard radar and searches for areas where time critical targets may be located. Upon detection, it releases one or more loitering area dominance munitions that are equipped to search an area and destroy targets of a certain type within that area. Fixed targets are detected from an airborne sensor with an infinite field of regard. This simulates the intelligence gathering process that occurs in the months leading up to the conflict. Perfect communications are assumed, although LRS systems may be plagued by sensor-to-shooter time delays and other lags defined by the rules of engagement. Potential targets include SAM sites, radars, and mobile time critical targets. Other high value targets and deeply buried hardened targets are included in the scenario.

5.3.1.3 Create Hierarchical Functional Architecture

The OV-1 concept graphic defines the general focus of the LRS architecture. The next step is to define the functions that this architecture must perform using a top-down, systems engineering decomposition. A functional architecture "identifies and structures the allocated functional and performance requirements" [130]. By decomposing requirements at the highest level into lower level functions, a product (system or system-of-systems) can be described "in terms of what it does logically and in terms of the performance required" [130]. The traditional functional decomposition process must be modified somewhat for the new capability-focus, as hard requirements are not always handed down from the acquisition authority [359]. Brainstorming of potential CONOPS at the highest level should define functions consistent with the overall capability. From Figure 42, a reasonable starting point for the functional decomposition in this research is with the operational activities in the fourth QFD matrix, known in the Air Force as the "kill chain."

It can be argued that the functions of warfighting have not changed since the dawn of armed conflict. A functional decomposition for a military system reveals two primary functions: *find* and *kill*. Traditionally, the more difficult of the two functions is find. In ancient times, armies marched great distances to engage enemy forces. Campfires, trampled trees, and rustling animals assisted the find function when remote sensing technologies were limited to the spyglass and mounted scouts [416].

Current military systems use a combination of satellites, manned and unmanned aircraft, special forces troops, and other human intelligence to find targets of interest. The success of the find function varies depending on terrain, weather, information sharing, and a myriad of other confounding factors. In the difficult urban warfare situations faced today, the find function is arguably the most difficult: potential terrorists look like ordinary citizens.

The endgame of combat is the kill function. Once the find function has been performed successfully, the kill function is comparatively easy. The above examples have obvious corollaries to how the kill function is executed. The combination of these two functions comprises the "kill chain" for military combat. As noted by Frits, every function has an associated probability of success (in this case, P_{find} and P_{kill}), and an associated time (T_{find} and T_{kill}) [159]. The probability of success of this kill chain can be found as the product of all the subjective probabilities for the functions defined.

$$P_{success} = P_{find} \times P_{kill} \tag{2}$$

While the characteristic time to complete the kill chain is given by the sum of the characteristic times to perform each function.

$$T_{operations} = T_{find} + T_{kill} \tag{3}$$

It is obvious that if the probability of any element in the kill chain is zero, the probability of success is zero. It is also important to note that T_{find} and T_{kill} may be on very different timescales[13].

The Air Force decomposes the kill chain into six elements: find, fix, track, target, engage, and assess (see Figure 44). The first four functions shown in blue can be loosely mapped to the *find* function described above whereas the last two describe the *kill* action.

[13]Special forces troops have been attempting to find Al Qaeda terrorist leaders for a number of years. If the find function were completed successfully, it could be argued that the kill function would occur in a dramatically smaller timeframe.

In 2005, the AFRL appended this mantra with a preceding "anticipate" and "anyone, anytime, anywhere" on the back end [371]. Abbreviated AF2T2EA4, this modified kill chain reflects the focus of current technology innovation efforts [229]. According to Brown, "The A's are where most of the capability gaps exist" [78]. This is primarily due to the uncertainty associated with these functions and the rapid pace with which enemies update their tactics to confound attempts to thwart them.

Figure 44: Functional Decomposition: The Kill Chain.

Anticipate, which means "to feel or realize beforehand," refers to the ability of Air Force leadership to predict when and where targets are present [22]. This function is greatly aided by intelligence, surveillance, and reconnaissance and relies on efficient battle management and communication with in-theater assets. The next function, *find*, is where the Air Force is currently devoting a majority of its technology and tactical development efforts [407]. Operation *Allied Force* relied heavily on the in-flight redirection of U-2 spy aircraft to areas of interest. Advanced multi-spectral sensors allowing visualization through ground clutter and sensor fusion with various battlefield assets are increasing the effectiveness and reducing the time of the find function. *Fixing* refers to "making an accurate determination of location" [407]. Today's precision weapons require very precise information for targeting. Fixing the location may involve laser designation or comparing real-time imagery to satellite photos to determine GPS coordinates of the target. *Target*, defined as "to aim at or for" is the act of calibrating the location information with the fix function to the asset that engages the target [22]. The *engage*[14] function, according to former Air Force Chief of Staff John Jumper, has always been the "strong suit" of the Air Force [407]. The Air Force has a variety of munitions to engage various targets including moving targets, combat aircraft, fixed emplacements, and hardened deeply buried targets, albeit with different levels

[14]Note that search and rescue teams can also "engage" a "target," although the functions are fulfilled by physical implements consistent with the search and rescue mission.

of effectiveness depending on the target type and the conditions under which the weapon is employed. The final function, *assess*, determines whether the engage function was successful and the target has been destroyed. UAV-mounted cameras have greatly increased the effectiveness of this function. One tactic to assist with battle damage assessment is to fire two weapons with nose-mounted cameras at a target. The first weapon strikes the target and the second one will photograph the target after the impact of the first weapon, although admittedly this doubles the cost of combat operations. The assess function can be challenging: Tirpak notes that in Operation *Allied Force*, on-site inspectors were required to assess whether or not Serbian tanks were destroyed by NATO bombings [407]. A controversy later erupted when the official NATO damage assessment overestimated Serbian claims by an order of magnitude [46]. Clearly, the assess function is critical to overall mission success, as it reveals whether or not a hostile asset has been sufficiently damaged to negate the need for retargeting.

Multiple systems are often employed to perform the functions of the kill chain. Gaps indicate opportunities for the target to escape, breaking the kill chain. In some cases, multiple systems work together to provide location information using a technique Jumper refers to as "Wolfpack ISR" [407]. While it is possible for a single system to perform all functions in the kill chain, a more likely implementation relies on fusing sensor data from multiple ISR assets, directing the appropriate asset(s) to engage the target, and assessing damage using multiple means. The integration of these functions to increase the probability of success and decrease the characteristic time to complete the links in the kill chain is the primary objective of *Network Centric Warfare* [23]. To aid in the mapping of system elements to functions, it is necessary to further decompose these operational activities into system functions.

5.3.1.4 Map Operational Activities to System Functions

The fourth QFD in Figure 42 relates the operational activities to the system functions that must be performed. In the kill chain, all operational activities must be performed; however, different combinations of system functions can lead to the completion of the kill chain depending on the circumstances of the scenario. For example, due to their high mobility, air targets are usually not detected from surveillance photos. Radar is the primary means of detection for air targets. The Find operational activity for air targets would therefore follow a sequence such as the ones shown in Table 3.

Table 3: Functional Decomposition of Find Function for Air and Ground Targets.

Air Targets	**Ground Targets**
Select Airspace Region	Select Region of Interest
Detect Targets	Gather Imagery
Classify Targets	Relay Imagery
Identify Targets	Gather Radar Data
Relay Probable Target Location	Relay Radar Data
	Fuse Sensor Data
	Detect Targets
	Classify Targets
	Identify Targets
	Relay Probable Target Location

The DoDAF Operational Activity to Systems Function Traceability Matrix (SV-5) "depicts the mapping of operational activities to system functions and thus identifies the transformation of an operational need into a purposeful action performed by a system" [137]. A generic SV-5 matrix is shown below in Figure 45. The DoDAF mapping is rotated 90° from the depiction shown in Figure 42 because the QFD formulation traditionally lists the "what's" on the left side and the "how's" across the top.

The SV-5 matrix for the Long Range Strike system architecture is shown in Figure 46. The functions in this matrix were determined by examining existing systems and the functions performed by them. It is important to note that these representative system functions are particular to the conceptual model for this research and are not necessarily the exact functions performed by Air Force assets. This canonical set includes the majority of the functions associated with supporting the Air Force Global Strike task force and defines the functions that must be performed by cognition models in the scenario [230]. Simulations over longer timescales requires cargo resupply, logistics, personnel management, health services, and other functions consistent with the "Agile Combat Support" core competency of the Air Force [438].

Operational Activities

System Functions	3.11	3.11.3	3.12	3.12.1	3.12.2	3.12.3	3.13	3.14	3.14.1	3.14.2	3.14.3	3.14.4	3.15	3.16	3.17	3.17.1
1	X															
1.1		X														
1.1.1			X													
1.1.1.1	X															
1.1.1.2					X											
1.1.1.3							X									
1.1.2										X						
1.1.2.1				X												
1.1.2.2						X										
1.1.2.3								X								
1.1.3											X					
1.1.3.1													X			
1.1.3.2									X							
1.1.3.3														X		
1.1.3.4														X		

Figure 45: DoDAF (SV-5) Operational Activity to Systems Function Traceability Matrix [137].

Operational Activities

System Functions	Find	Fix	Track	Target	Engage	Assess
Gather Images	X					
Process Images	X					
Detect Targets	X					
Determine Environment	X					
Classify Targets	X					
Identify Targets	X					
Fuse Sensor Data	X	X				
Assess Target		X				
Track Until Stopped			X			
Geolocate Target			X			
Update Target List				X		
Assess Engagement Capability				X		
Relay Target Coordinates				X		
Assign Targets				X		
Plan Route(s)				X	X	
Execute Force Order					X	
Support Weapon Flyout					X	
Lock Weapon onto Target					X	
Destroy/Degrade Target					X	
Collect Battle Damage Information						X
Assess Battle Damage Information						X
Relay Target Status to Battle Mgr						X

Figure 46: DoDAF (SV-5) Matrix for the strike mission of an LRS Architecture.

5.3.1.5 *Brainstorming to Find Potential System Elements of an LRS Architecture*

Once the functions for the LRS architecture has been determined from the aforementioned functional decomposition, the next step is to identify types of systems that comprise the LRS architecture so that physics-based models of their performance can be defined. A literature search of public documents, Air Force posture statements, transformational plans, and fact sheets was conducted to identify potential elements of an LRS architecture. This was supplemented with a brainstorming activity as shown in Figure 47. The mind map brainstorming technique was used with LRS capability as the central focus, systems, software, missions, support capabilities, doctrine, tactics, and technologies can all be connected to this link: the mind map contains *anything* that is of interest for LRS capability. Several of the branches are expanded to show detail. Several links are also provided, for example, noting that weapons are subsystems and they require logistics for their resupply. This depiction was created using the OpenMind software, which allows files, pictures, video, and other documents to be attached to each item in the mind map [273]. In a collaborative design activity, team members can use it to link to a repository of information, which then serves as a knowledge base for new team members. For example, photographs and specification sheets for each of the existing military systems in the architecture are linked to this mind map.

This activity is also useful in identifying elements of the current and proposed military architecture which a new asset must interoperate with. For example, KC-135 *Stratotanker* and KC-10 *Extender* aircraft utilize a boom for refueling operations. The maximum fuel transfer rate through the boom is 7,524 lb/min [441]. This constraint limits the maximum fuel transfer rate of a future LRS aircraft unless a new tanker is part of the architecture under consideration. Examination of the current Air Force systems and a review of pending development programs reveals that a majority of Air Force assets will not receive extensive upgrades or replacement over the next several years. Notable exceptions include a proposed

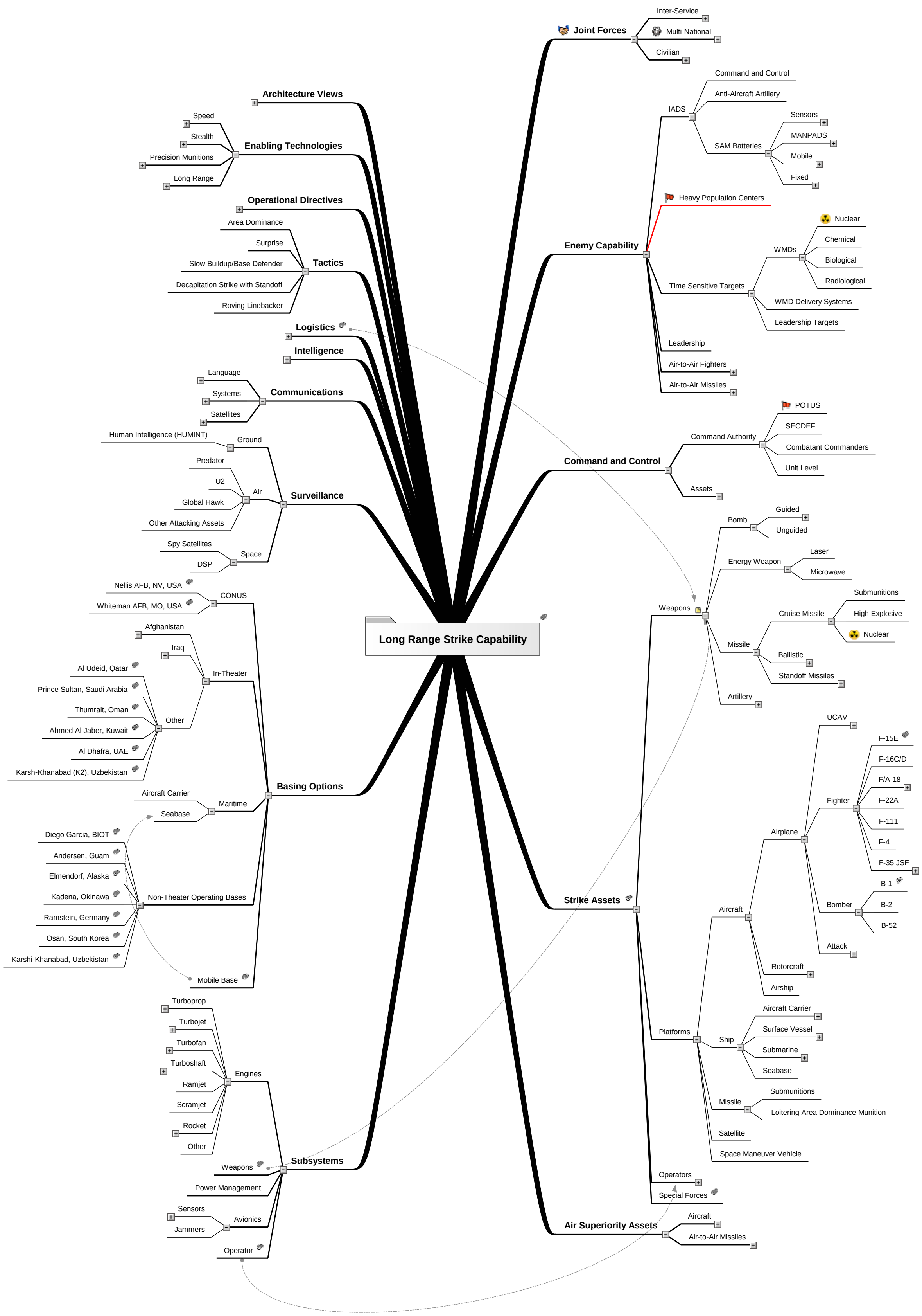


Figure 47: Mind Map Brainstorming Exercise for LRS Capability.

Multi-Sensor Command and Control Aircraft, the E-10A [376] to replace E-3 *Sentry* and E-8C JSTARS aircraft[15], and the gradual replacement of the A-10 and F-16 fleet with F-35A fighters. The specific systems selected for the LRS architecture and the reasons therefore are summarized in a subsequent section.

5.3.1.6 Map System Functions to Physical Systems Using a Matrix of Alternatives

A large part of architecture development is design for interoperability with existing assets. While a LRS architecture is comprised of many support assets that perform functions such as air superiority, SEAD/DEAD, combat support, jamming, and the like, the focus of this step is the identification of the physical systems required to perform the system functions identified in Figure 46. The final QFD is the decomposition process introduced in Figure 42 is a matrix of alternatives that relates system functions to physical systems. In 2005, Engler developed an Interactive Reconfigurable Matrix of Alternatives (IRMA) to decompose a Long Range Strike asset within the physical domain while tracking the dependencies between multiple rows [144]. This technique can be extended to list system functions as the **rows** of the IRMA and system options as the **columns**. Each row should include an option for "other," leaving room for infusion of new systems at any point in the architecture. Also, rows can be left blank, indicating that some system functions identified in the "as-is" military may be eliminated in the proposed LRS system architecture as technologies are infused and capabilities are provided by different systems. The matrix of alternatives for the modeled Long Range Strike architecture is shown in Figure 48. Finally, it is interesting to note that over 3.54 x 10^{25} combinations of system architectures can be composed from this matrix of alternatives. This large number results from the fact that many selections in the matrix of alternatives are non-unique, as multiple elements from the same row can be synthesized to perform a given function.

It is also of interest to specify a matrix of alternatives to define the adversary in the simulation. The options in the matrix of enemy alternatives (over 4.0 x 10^{18} combinations)

[15]Which, according to the most recent QDR, may be replaced by a constellation of Space Based Radar assets [456].

encompass various degrees of threats from a Soviet-like enemy to a rogue state or non-state actor terrorist group. For this testbed demonstration, it is desired to calibrate the enemy architecture based on the parameters of the Iraq scenario presented in Operation *Desert Storm*. Section 5.2 details the specific characteristics of the enemy scenario, which is summarized in Figure 49. Notable differences from the actual situation include the selection of generic SAM sites to avoid sensitivity issues with the identification of specific threat systems and the removal of air-to-air combat assets from both the hostile arsenal and the order of battle of friendly forces (Figure 48).

The green highlighted boxes in the matrix of alternatives show decisions that have been made to date based on the goals of the methodology, the up-front motivation for the study, and a literature search of available bases, systems, and architecture elements. Traditionally, the number of elements in a matrix of alternatives can be calculated by multiplying the number of options in each row times the number of options in every other row, as shown in Equation 4.

$$\#Alternatives = \prod_{Rows} OptionsPerRow \tag{4}$$

The definition of system-of-systems architectures is more complex. In most cases, the rows have the property of non-exclusivity, which means that each row does not have a single unique solution. As a result, the number of combinations in each row is not additive, it is **combinatorial**! The number of alternatives defined by a matrix of alternatives that has more than one valid option per row is given by:

$$\#Alternatives = \prod_{Rows} \left(2^{Non-ExclusiveOptionsPerRow} + ExclusiveOptionsPerRow\right) \tag{5}$$

Through the use of intelligent filters and parametric decomposition of the space, the number of combinations can be greatly reduced. For example, if a fighter aircraft in Figure 49 has parametric properties that range between a Generation 2 and Generation 4+ fighter, then the "air-to-air assets" row decreases from 513 combinations to 1. In the spirit of Thoreau, "simplify, simplify, simplify!" is the guiding mantra of the architecture setup [405]. The matrix of alternatives in Figure 48 explains this simplification.

To summarize the major downselections in Figure 48, although a variety of overhead

Physical Systems

System Functions

Gather Images	Predator	U-2	Global Hawk	Airship	Milsat	Commercial	Other
Process Images	NGA	DCGS	JIOC				
Detect Targets	JSTARS	MC2A	Sensorcraft	PISR	SBR	Global Hawk	Other
Locate Targets	CAOC	DCGS	JIOC	MC2A	PISR	SBR	Other
Classify Targets	CAOC	DCGS	JIOC	MC2A	Self		
Identify Targets	CAOC	DCGS	JIOC	MC2A	Self		
Fuse Sensor Data	CAOC	DCGS	JIOC	MC2A	Self		
Locate Target Coordinates	GPS	Other					
Track Ground Targets	JSTARS	MC2A	Self	Other			
Relay Target Coordinates	MILSTAR	DSCS	FLTSATCOM	UFO	SDS	Iridium	Other
Assign Targets	CAOC	DCGS	JIOC	MC2A	Self		
Develop Track to Target	CAOC	DCGS	JIOC	MC2A	Self		
Relay Target Track	MILSTAR	DSCS	FLTSATCOM	UFO	SDS	Iridium	Other
Obtain Permission to Attack Target	NCA	Cmdr.	Self				
Fly to Target (LRS Asset)	Aircraft	Spaceplane	Missile	Satellite	Airship	Helicopter	Other
Lock Weapon onto Target	Radar	Laser	Visual	GPS			
Fire Weapon	Energy	Bomb	Missile	Gun	Self	Submunition	Other
Observe Attacked Target	Predator	U-2	Global Hawk	Airship	Milsat	Self	Other
Determine Target Status	Predator	U-2	Global Hawk	Airship	Milsat	Self	Other
Relay Target Status to Battle Mgr	MILSTAR	DSCS	FLTSATCOM	UFO	SDS	Iridium	Other

Figure 48: Matrix of Alternatives: Mapping of System Functions to Physical Systems.

Enemy Attributes	Theater of Interest	Western Asia	Northeast Asia	Eastern Europe	CONUS	Other
	Enemy Allies Involved	Yes	No			
	Nearby Friendly Bases Available	Many	Some	Few	None	
	Surface-to-Air Missiles	SA-2	SA-3	SA-5	SA-8	SA-XX
		Other Soviet	Roland	MANPADS	Generic	Other
	Anti-Aircraft Guns	ZSU-series	Other	None		
	Air-to-Air Assets	MiG-29	MiG-31	SU-27	F-15	UCAV
		Gen 2 Fighter	Other Gen 3	Gen 4+ Fighter	Other	None
	TBM Launcher Technology	None	SCUD-B	Modified SCUD	Extended Range	Other
	Fixed Base/Airfield Density	None	Low	Medium	High	Unknown
	Surface-to-Air Defense Density	None	Low	Medium	High	Unknown
	Air-to-Air Asset Density	None	Low	Medium	High	Unknown
	High-Value Fixed Targets	None	Low	Medium	High	Unknown
	High-Value Moving Targets	None	Some	Few	Many	Unknown
	High-Value Target Types	WMD Facilities	WMD Launchers	Leadership	C&C	Other
	NBC Capability	None	Limited	Extensive	Ubiquitous	Unknown
	Deeply Buried Target Prevalence	None	Some	Few	Many	Unknown
	C4ISR Capability	Integrated	Distributed	Localized	Ad-Hoc	None
	Willingness to Attack Civilians	None	Low	Medium	High	Unknown
	Willingness to Use WMD	None	Low	Medium	High	Unknown

Figure 49: Matrix of Alternatives for Hostile Elements.

imagery assets are available, targets are detected and identified by either an overhead satellite in the case of fixed targets or an airborne sensor like a Predator, Global Hawk, or U-2. To simplify the command and control process, the Distributed Common Ground System (DCGS) gathers target information and parcel it out to the Combined Air Operations Center (CAOC) to task individual assets. Communication between these entities and strike aircraft uses a communications system based on the MILSTAR satellite communications system and the Link-16 protocol, assuming that a MILSTAR satellite is always overhead and sufficient bandwidth is available to support transmission of targeting information. In the simulation, the rules of engagement allows individual aircraft to select and attack their own targets, eliminating the need to receive engagement orders from a national command authority or the battlefield commander for each target. It is assumed that once a target has been assigned by the battle manager, engagement permission is implicit. As mentioned in Section 2.4.3, even though many different assets are proposed for the LRS mission, this research focuses exclusively on aircraft-based solutions. Munitions will be limited to bombs and missiles, although both are assumed to be guided projectiles. Battle damage assessment (BDA) is conducted by the same overhead assets that detected and identified targets, and the same MILSTAR communication setup is used to transmit battle damage information back to the battle manager.

5.3.1.7 Identify Models and Linkages Required for Scenario Development Using the SysML

In a system-of-systems, "everything impacts everything." Elements of the system-of-systems and the linkages between them will impact an architecture. The first desire of a systems engineer is to model all of these elements and their interactions. For a system with n elements, the number of models is:

$$N_{models} = n_{elements} + 2^{n}_{connections} \tag{6}$$

For a 20 element system, the number of models required for all elements and connections is *only* 1,048,596. In reality, everything does not impact everything *directly.*

As Ilachinski notes, the reductionist philosophy of the Western scientific method that decomposes a system into smaller pieces causes the emergent properties of a system to be lost: “In the act of exploring properties reductionism loses sight of the dynamics. The analysis of complex systems instead requires a holistic, or constructionist, approach” [214]; however, although a system-of-systems relies on these interactions to derive its emergent behavior, the required connectivity between systems should be defined by the *sensitivity of MoEs to the system connections*. The previously mentioned techniques of functional decomposition and matrices of alternatives narrow the simulation space by focusing on a canonical set of models and the ANOVA technique can be used to statistically determine the sensitivities.

A technique from the software engineering field, the Systems Modeling Language, will refine this space even further. The SysML, detailed in Section C.5.1, can be used to quickly diagram objects that are modeled in the simulation framework [388]. This technique is useful for diagramming existing code and planning the conceptual model of a modeling and simulation environment. It is important to note that while, in general, the simulation approach and proposed methodology are framework independent, after the conceptual model is created the simulation becomes somewhat dependent on the framework. This is because specific models must eventually be coded and linked within a simulation software tool. Choices such as the coding language, variable names, interface types, and physics-based algorithms must eventually conform to framework-specific standards, but the SysML diagrams that describe the conceptual model are independent of the framework choice.

One SysML depiction, an activity diagram, “shows a sequential flow of actions” and “is typically used to describe the activities performed in a general process workflow” [145]. An activity diagram can also be used to depict the flow of information between various cognition models that make up the kill chain as shown in Figure 50. The colors indicate the six central functions of the Air Force kill chain (further explained in Section 5.3.1.3). Several cognition models including the GITBattleManager, the GITFCGroundController, the GITBomber, and the FLAMES example model FQWASFire[16] are used to perform ground attack functions for the LRS. The functions listed are consistent with those described

[16]The weapon firing routine within the air-to-surface weapon system.

in the SV-5 matrix in Figure 46 and the matrix of alternatives in Figure 48.

Model development was required across all of these cognition models to provide realistic cognition effects in the testbed simulation. The development of each of these models is described in Section 5.3.2.1. Model development could include creation of new models from scratch, modification of existing models to fix errors, or modification of the fidelity level of existing models.

Various SysML diagrams were used throughout the model building phase to identify model interfaces, specify module functionality, and maximize the reusability of developed models for other simulation activities. Once the scope of the simulation has been reduced to "the leanest physical experiment that would result in satisfactory answers for the questions," using the aforementioned techniques, the next step in the process is to begin to construct a computational simulation that is consistent with the conceptual model [494].

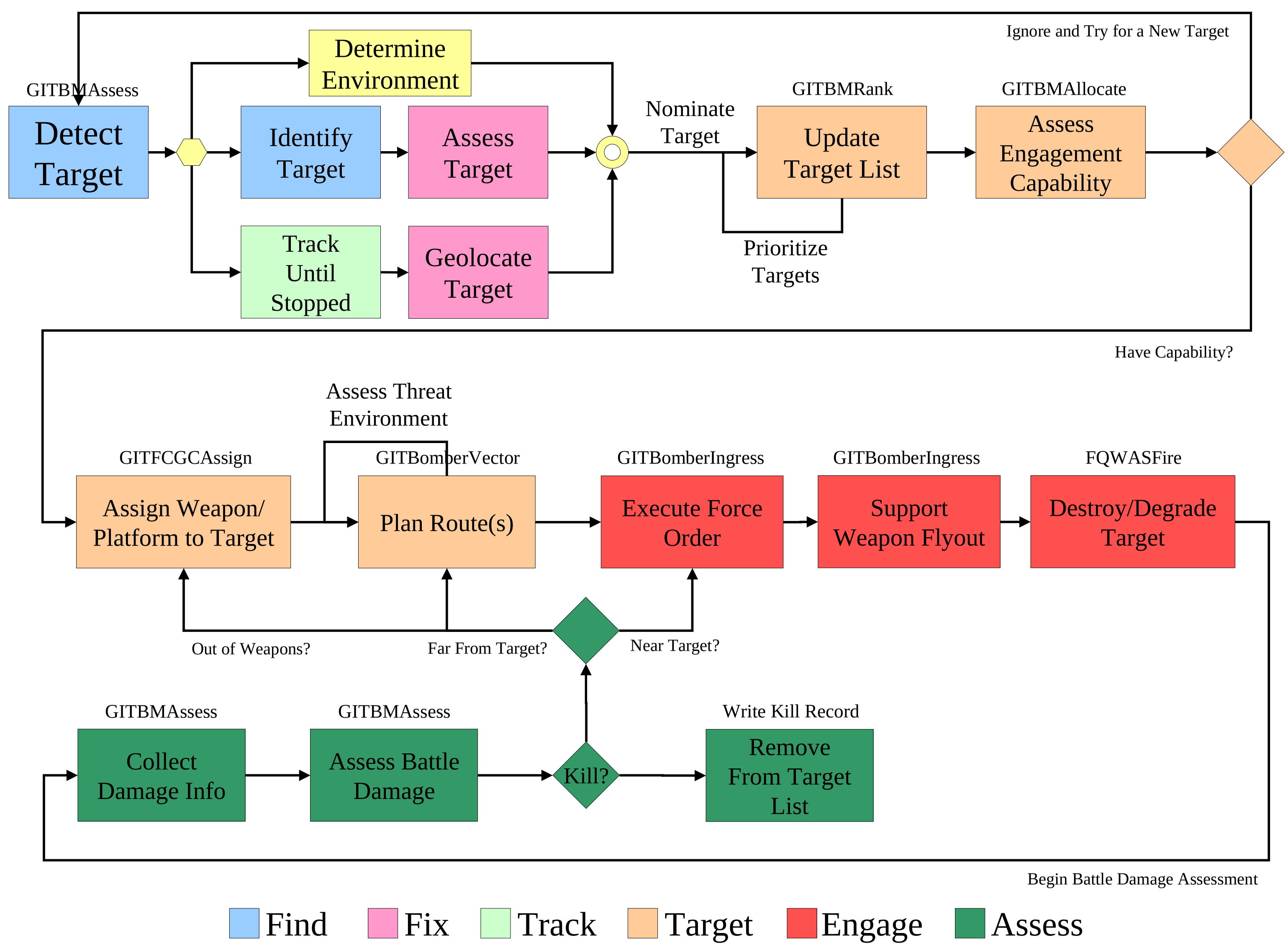


Figure 50: Activity Diagram for Strike (Adapted from Reference [134]).

5.3.2 Step 3.2: Develop Representative Models to Evaluate Technologies

> *"By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work."*
>
> -John Von Neumann

After the learning curve associated with the computational framework has been overcome, the next step is the creation of a *representative* testbed scenario that examines LRS in a system-of-systems context.

5.3.2.1 Development of Models to Support Scenario

Creating an object oriented simulation requires a myriad of models to represent the behavior of elements within the system-of-systems. Central to this desire is the creation of validated models, that is, models that are syntactically correct[17]. Models in the FLAMES framework are of two primary types:

- Physics models, which describe the behavior of entities with respect to physical laws.
- Cognition models, which define the actions that an entity performs, emulating the behavior of the real system.

Section 5.3.1.6 described a process that uses a matrix of alternatives to identify which physics models need to be created to support the LRS development effort and Section 5.3.1.7 showed an example of how the SysML is used to specify cognition models and their linkages. The validation of physics models is fairly straightforward. FLAMES example models, created to showcase the abilities of the framework, have several notable errors in their physics which are often easily corrected with several lines of code. The cognition models, on the other hand, require far more development to produce a realistic simulation where the agents perform functions as they would in the "real world." According to Alberts and Hayes, "Models are often a mix of what we know (or think we know) and what we

[17] Verification is "doing the right thing" and validation is "doing it right."

think (conjecture or hypothesize)" [41]. The verification and validation of cognition models is therefore an iterative process: observing the behavior of the model provides the designer with more information that can be used to tune the model. The first step in developing robust cognition models for the simulation environment is to identify the primary cognition models that are needed to realistically simulate a LRS system architecture.

There are two general ways to develop cognition models in FLAMES. First, the FLAMES code generator can be used to create empty models with the appropriate linkages to support model development from scratch; however, the learning curve on this method of model development is very high. An approach recommended by Ternion is the modification of several example models shipped with the software to suit the needs of the user.

5.3.2.2 Identifying Example Cognition Models for Modification

There are several rudimentary cognition models that are shipped with FLAMES to demonstrate the ability to perform a defensive counter air mission. While this mission differs greatly from the parameters required for an LRS scenario, the logic in some of the example models can be modified to support LRS modeling. Example cognition models of interest include:

- FCSAMController (fcsc): Air defense controller that detects airborne targets, calculates track information on the targets, and conveys the track information to a fighter controller.
- FCAirController (fcac): Fighter controller that receives track information on hostile fighters and assigns air-to-air fighters to attack the identified targets.
- FCFighter (fcftr): Air-to-air fighter cognition model that receives track information about hostile fighters or bombers and is vectored near them. The FCFighter then begins to search for hostile targets in that area, engages targets, and returns to a patrol after engaging targets.
- FCAirInterdiction (fcain): Air-to-ground attack aircraft that can be vectored toward a geographic feature, a coordinate, or a unit. The FCAirInterdiction entity attacks only one target that is defined in the script of the scenario.

In the command hierarchy for the models defined above, the commander of an FCFighter is an FCAirController, and the commander of the FCAirController is an FCSAMController. The FCSAMController is capable of assigning either fighters or SAMs to attack hostile air targets. Unfortunately, the FCAirInterdiction cognition model is very simple and cannot be coupled to any of the other cognition models.

5.3.2.3 Modification of Cognition Models to Create an Intelligent Battle Manager

One of the stated objectives of the proposed methodology is the development of an *intelligent battle manager* that has some knowledge of strategic and tactical decisions to eliminate the need for a trained operator to observe individual case runs and make tedious platform/weapon allocations. The creation of the intelligent battle manager partially relies on the development of cognition models that support realistic processes consistent with the activity diagram in Figure 50.

First, the example models are copied and their properties are altered to perform new functions. The FCAirController model was copied and defined as a GITFCGroundController[18]. The goal of this cognition model is to allocate *ground* targets (as opposed to air targets) for assignment to friendly LRS aircraft.

Next, since the ground controller model does not have a method of detecting, identifying, and prioritizing targets, it was necessary to copy the SAM controller model and define it as a GITBattleManager. The FCSAMController model is actually an air defense controller capable of directing both SAM sites and fighters to perform a Defensive Counter Air (DCA) mission. The battle manager has the same functions as a SAM controller, except that it only identifies ground targets. A combination of GITBattleManagers with subordinate GITFCGroundControllers and FCSAMControllers with FCAirControllers can be used to provide both ground and air coverage of the battlespace.

The next logical step would be to modify the fighter cognition model to pursue ground

[18]The prefix GIT stands for "Georgia Institute of Technology" and was defined by Ternion to identify models developed at GIT. The prefix BOE is used by Boeing, VAC is used by the Vehicles Directorate at AFRL, and the prefix GTI is used by the National Air and Space Intelligence Center (NASIC).

targets instead of air targets and maintain the same command chain, communication protocols, and subordinate registration methods between the originally defined command chain; however, the logic in the fighter model is "too smart." In the default model, a fighter is vectored *near* an enemy since air targets move frequently from place to place. Once in the area, the fighter begins to look for a viable target. When this cognition model is applied to ground targets, the fighter attacks any target on the ground that is near its current position, often ignoring the target that it was vectored to attack. While this cognition model is useful for pursuit of mobile targets, an alternate approach to attacking fixed ground targets is needed.

As an alternative, the relatively "dumb" FCAirInterdiction could be upgraded to receive commands from a ground controller. The FCAirInterdiction model was copied to create a GITBomber cognition model. After creating the necessary message models and registration protocols, GITBombers can be assigned targets from a GITFCGroundController (which in turn receives targets from the GITBattleManager). Unfortunately, the cognition logic in the GITBomber model does not support multiple targets assigned to the same bomber. The bomber only attacks the last target in its prioritized list. As a result, a complicated target management routine was created that feeds each bomber only a single target. After completing the engagement with one target, the bomber again becomes available for tasking and can be assigned targets from its ground controller.

The air-to-surface weapon system used by the bomber did not have a method to account for the lack of availability of the weapon system once all munitions were expended. A query method was written to ensure that only GITBombers with available weapons were tasked to an engagement.

If the weapon system is unavailable (all munitions expended), the GITBomber is vectored toward its commander, located at its home airfield. This required modification of the "MOVE TO LOCATION" function to accept altitude and speed inputs in addition to the location value. When within 10 km of the airfield, the GITBomber executes the "LAND" function of the GITFQPFixedWingPlatform which brings the platform to minimum airspeed and zero altitude. After a user-defined time period, the GITBomber is rearmed and the maximum assignments is incremented by one so that the battle manager recognizes this platform as an active unit available for tasking.

Section B.1 defined a method for prioritizing targets within the battle management routine. Applying this technique to the FLAMES models required extensive model development. First, a majority of the strategic targets in Operation *Desert Storm* were fixed facilities. A new platform model called GITFQPFacility was derived from the ground vehicle model and implemented in FLAMES. The ground vehicle model was used so that "facilities" could be mobile or stationary. Several of the inherent properties of the vehicle model were also of interest for the facility model, for example, the properties of "vehicle length" and "vehicle width" can be used to store runway length and width for facilities representing airfields. An example of the platforms window showing several platforms of the GITFQPFacility class is shown in Figure 51.

Next, a method was needed to assign the properties of the "threat value" to each platform. An approach was developed that utilizes a powerful function in the FLAMES kernel. All platforms (and for that matter munitions) in FLAMES can be assigned a *signature*, a generic property that is visible to a sensor. Signatures are defined in terms of a radar cross section (m^2) and a lower and upper frequency bound (MHz). Since the signature property can be attached to any properly defined entity in the simulation, a series of *pseudo-signatures* were created to emulate the threat value. It is important to note that these signatures are not real RCS values that describe the visibility or vulnerability of the platform and are only a simplified way to assign properties to any entity in the simulation.

Four levels of threat intensity were defined for each target set with the severity represented by a RCS of 1, 3, 9, or 81 m^2. A false frequency between 7000 and 7055 MHz at increments of 5 MHz was used to define the threat level using the FGroundRCS attribute of the signature. The defined frequency bounds are shown in Table 4.

For each platform, the "signature selector" window, shown in Figure 52, is used to assign physical signatures and pseudo-signatures to a given platform. In the example shown, an airfield can be assigned a Ground RCS and an Air RCS so that it can be detected by both air and ground sensors, as well as several threat signatures representing its characteristics

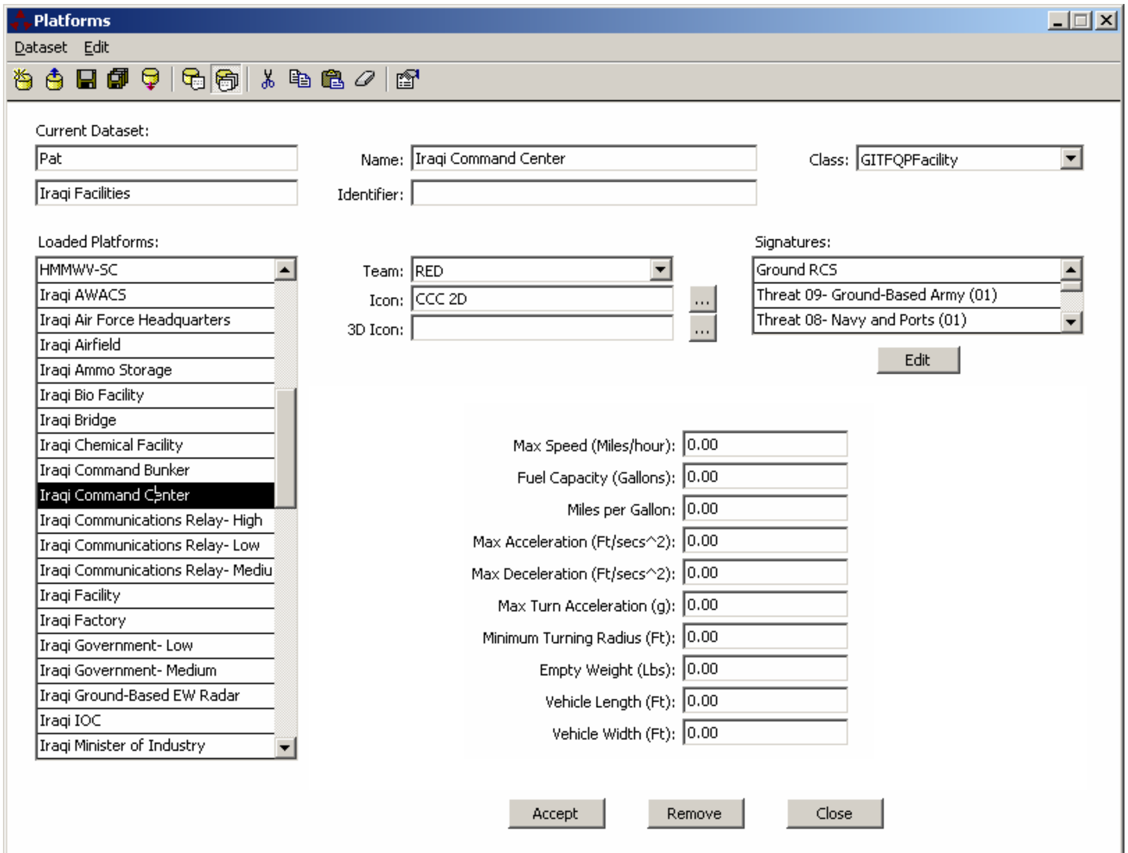


Figure 51: The Platforms Window in FLAMES, Showing a GITFQPFacility.

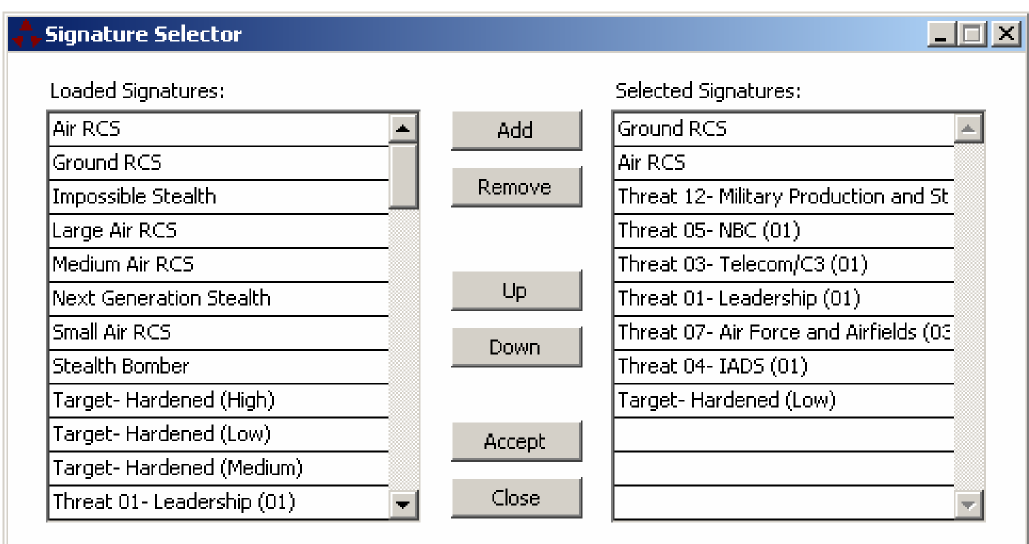


Figure 52: Using the Signature Selector Window to Assign Pseudo-Signatures to a Platform.

Table 4: Pseudo-Signatures Used to Prioritize Strategic Targets (MHz).

Target Set	Frequency	Target Set	Frequency
Leadership/Command	7000	Air Force and Airfields	7030
Electricity	7005	Navy and Ports	7035
Telecommunications/C3	7010	Army	7040
IADS	7015	Oil Production/Storage	7045
Nuclear, Chem and Bio	7020	Roads and Bridges	7050
TBM Launchers/Prod.	7025	Military Storage/Prod.	7055

in the target sets of (12) military production and storage, (5) NBC, (3) telecom and C^3, (1) leadership, (7) air force and airfields, and (4) IADS. In the example shown in Figure 52, all of the signatures used have an RCS of 1.0 except the air force and airfield signature which has a medium value of 3.0. Using this mix-and-match approach, platforms can be created which have threat characteristics of varying levels across all target sets.

This setup uses 48 discrete signature options (4 levels for each of 12 target sets) to define the threat level of a target. Continuous values cannot be used for individual targets because the FLAMES kernel does not support a method for altering an existing RCS. Doing so would require defining a signature and a platform element for each target in the scenario, which is inconsistent with the object-oriented paradigm of inheritance. An alternative method would be to create signatures in each target set in increments of 1, 2, 4, 8, 16, 32, and the like. By converting a desired continuous value to binary and adding the appropriate discrete signatures using the signature selector to reach the desired value, finer discrimination between the threat levels of individual targets can be achieved. That level of resolution was not required for this work.

At runtime, the FPlatformGetSignature command in the FLAMES kernel can be used to retrieve the RCS in a given frequency band. To calculate the overall threat value of the target, this function is used twelve times across the band of pseudo-signatures defined in Table 4. The aggregated priority of target, T, is calculated using the equation:

$$(\text{Priority})_T = \sum_{j=1}^{12} (\text{ThreatValue})_{Tj} \, (\text{TargetSetImportance})_j \tag{7}$$

Where the Target Set Importance is defined by the QFD exercise as summarized in Section B.1. Equation 7 is used within the "RANK" function of the GITBattleManager to rank the list of detected targets based on priority. After ranking the target list, the battle manager then assigns the highest priority targets to platforms that are under the control of its subordinates. The GITBombers receive a target from the Ground Controller, ingress to the target, release a weapon, and either indicate their availability to receive a new target or return to base.

The aforementioned cognition models primarily address the issue of identifying targets, prioritizing targets, and tasking subordinates to attack the identified targets. The next major shortcoming that must be remedied is the fact that platforms are assigned in order of availability: the battle manager assigns the "next" platform in the list of available aggressors to attack the next most important target in the target list with no regard for the abilities of the platform or its associated munitions. This issue is addressed in detail in Section 5.5 which describes how the battle manager is trained.

5.3.2.4 Improving the Visualization of the FLASH Scenario Output

The FLASH two-dimensional viewer is useful for debugging both the cognition and physics of the developed models to ensure that the assets in the simulation exhibit realistic behavior. To aid in the analysis of the generated models, a routine was written to change the color of assets based on their state:

- Red, Blue, and White units represent hostile, friendly, and neutral units (FLAMES default)
- Black units are units detected by the battle manager. If a target cannot be seen by the battle manager, it retains the red color.
- Magenta units are hostile units currently being engaged.
- Yellow units are friendly units currently being tasked for an engagement.
- Green units are friendly units that have expended all munitions and are returning to base.
- After landing at the base, while idle the friendly units return to their original blue color.

Additionally, background images of the Earth, the Arabian Peninsula, Iraq, and Baghdad were imported as GeoTIFF files. The TIFF file format is a versatile raster data format used by the digital imaging community. This file can encapsulate metadata through "tags" within the image [351]. "GeoTIFF refers to TIFF files which have geographic (or cartographic) data embedded as tags within the TIFF file. The geographic data can then be used to position the image in the correct location and geometry on the screen of a geographic information display" [364]. JPEG files from the National Geospatial-Intelligence Agency (NGA) were converted to TIFF format and metadata tags for the correct geospatial information were added through an iterative process that matched the graphical image to known coordinates in the FLAMES scenario. This addition allows rapid identification of major geographic features such as borders, rivers, and lakes.

5.3.3 Step 3.3: Integrate Models to Create a Holistic Simulation

The third element of step 3 is the integration of the models described in Section 5.3.2 and the simulation elements detailed in Section 5.2 within the simulation framework. FLAMES requires the definition of platform elements that use physics models, dictionary entries that synthesize specific pieces of equipment from platforms, sensors, and munitions, and unit descriptions that specify the properties of individual elements in the simulation. Validation is conducted by attempting to calibrate the parameters of the existing models to match known outcomes of the initial sorties of Operation *Desert Storm*. In an example of one such test, appendix E.2 describes the calibration of the IADS model used in the simulation.

This integrated simulation forms the basic "game board" upon which technology evaluation studies are performed. As a final note, up to this point, all steps are required to perform a single simulation and generate a single data point. Subsequent sections describe how this simulation environment or its equivalent can be used to examine large amounts of data to discover valuable solutions throughout the design space. Infusion of new techniques and methods to provide output useful for capability planning and technology is also discussed throughout the next several sections.

5.4 *Step 4: Map Strategic Objectives to Actionable Operations*

> *"No one starts a war-or rather, no one in his senses ought to do so-without first being clear in his mind what he intends to achieve by that war and how he intends to conduct it."*
>
> -Carl Von Clausewitz,
>
> On War [100]

One of the primary technical challenges in creating an intelligent battle manager is how to prioritize targets for attack. In order to create a software solution to model real-world behavior, it is first necessary to define how the targeting, weaponeering, and battle damage assessment (BDA) functions occur in reality. The USAF targeting guide defines a process (see Figure 53) that can be used as a model of the interaction between the battle manager, strike aircraft, and battle damage assessment assets.

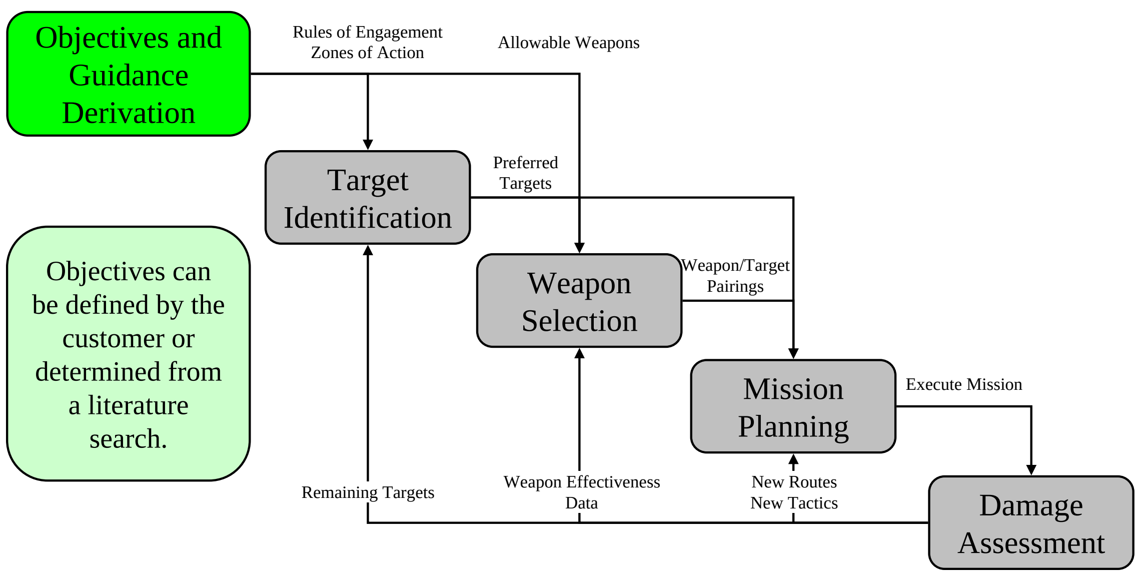


Figure 53: Cognition Model for Scenario Execution: The Target Cycle (Adapted from Reference [429]).

The first step, "Objectives and Guidance Derivation" identifies "what is to be achieved and under what conditions and parameters" [429]. A top-level objective such as "win the war" is realized through the satisfaction of multiple strategic objectives. Consistent with the GSTF mission, the primary objective of the LRS scenario is to clear the way for other friendly forces while neutralizing leadership targets, C3, IADS, and NBC elements in the opening stages of the conflict. The objectives and guidance for the Persian Gulf War used in the testbed activity are based on the National Policy Objectives specified in the U.S. Commander-in-Chief, Central Command (USCINCCENT) mission statement [457]:

- Neutralize Iraqi National Command Authority
- Eject Iraqi Armed Forces from Kuwait
- Destroy the Republican Guard
- Destroy Iraq's Ballistic Missile and NBC Capability
- Assist in the Restoration of the Legitimate Gov't of Kuwait

This list can be supplemented with some additional strategic objectives which are implied in the above and inferred from the priorities noted by Horner [98]:

- Obtain and Maintain Air Supremacy
- Cut Supply lines to the Kuwaiti Theater of Operations
- Destroy Saddam's Capability to Threaten Neighbors

These statements clearly define what needs to be achieved. The conditions and parameters include qualifiers such as "as quickly as possible" and "with minimum civilian casualties" that must be mathematically related to the MoEs and constraints defined for the scenario of interest.

The correlation of strategic objectives to functional target sets is consistent with Air Force doctrine: "airmen view the application of force more from a functional than geographic standpoint and classify targets by the effect their destruction has on the enemy rather than where the targets are physically located" [430]. While the locations determined in Section 5.2.1.3 define the operational constraints for aggressor aircraft, the functional and physical characteristics of a target will delineate its importance. A method for prioritizing targets with respect to the objectives and guidance is defined in the next section.

5.4.1 Target Development

The next step in the target cycle is target development, and is simplified to include only detection and identification of targets. It can be said that the target identification function seeks to identify the Clausewitzian "centers of gravity" within each target set. Additionally, "every target has distinct inherent, acquired, functional, physical, environmental, and mobility characteristics" as defined by the USAF Intelligence Targeting Guide [429]. The purpose of intelligence is to determine as many of these characteristics as are necessary for accurate targeting. Sensor assets in the simulation are tasked with detecting targets and relaying information on one or more characteristics to the battle manager. When sufficient information has been obtained to make a targeting decision, the battle manager assigns weapons to assets and construct an attack plan called the Air Tasking Order (ATO) which defines strike packages and time-over-target for thousands of aircraft [98].

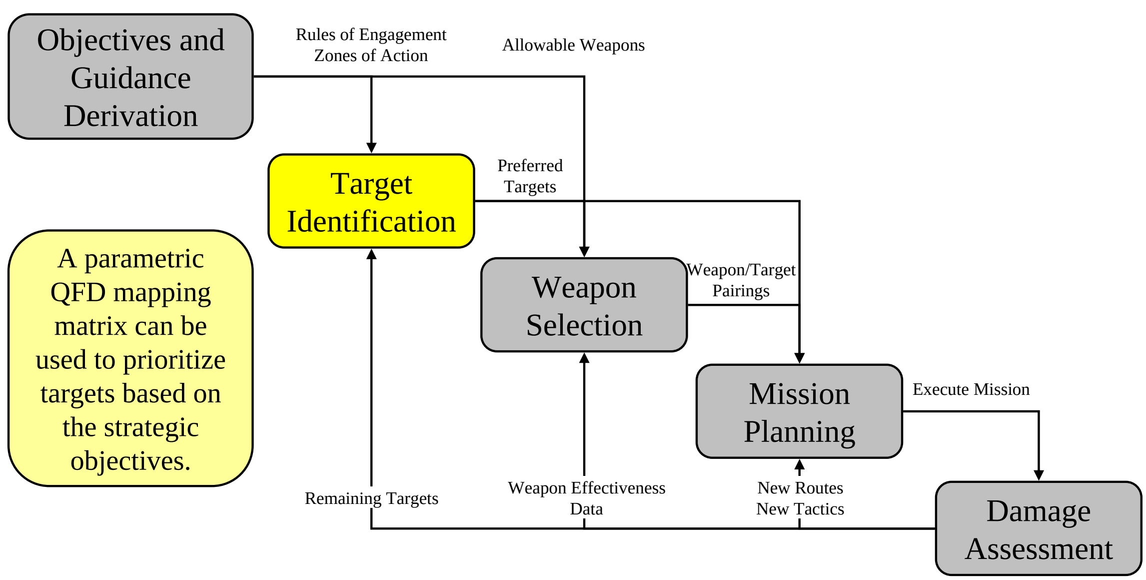


Figure 54: The Role of Target Identification in the Target Cycle (Adapted from Reference [429]).

Top-level leadership prioritizes military objectives in terms of the strategic guidance in the previous section; however, targeting experts need a means to map the functional properties of potential targets to the strategic objectives. Quality Function Deployment (QFD) is a popular approach in systems engineering that can be adapted for this purpose.

The mechanics of this technique are summarized in Section B.1. In this case, the "voice of the customer" maps directly to the strategic objectives (WHAT military planners want to do) and the "engineering characteristics" are in fact strategic target sets that must be struck to accomplish those objectives (HOW the objectives are accomplished). A QFD matrix that shows these relationships is depicted in Figure 55.

⦿ Strong Relationship
○ Medium Relationship
△ Weak Relationship

		Importance	Leadership	Electricity	Telecom/C3	IADS	NBC	TBM Launcher	Air Force	Navy	Army	Oil	Roads/Bridges	Mil. Storage
National Policy Objectives (CINCCENT Mission Statement)	Neutralize Iraqi National Command Authority	10	⦿	○	⦿	△	○	○	△	△	○			
	Eject Iraqi Armed Forces from Kuwait	0			○					○	⦿	○		⦿
	Destroy the Republican Guard	0	○	△							⦿			⦿
	Destroy Iraq's Ballistic Missile and NBC Capability	5	○	△			⦿	⦿	△	△	△			△
	Assist in the Restoration of the Legitimate Gov't of Kuwait	0	△								△			
Other Strategic Objectives	Obtain and Maintain Air Supremacy	8	△		○	⦿			⦿					△
	Cut Supply Lines to the KTO	0	○	○	○						⦿	⦿	⦿	⦿
	Destroy Saddam's Capability to Threaten Neighbors	2.5	⦿	○	⦿	⦿	⦿	⦿	⦿	△	⦿	○	○	○
Relative Importance			16%	5%	16%	13%	12%	12%	13%	2%	7%	1%	1%	2%

Figure 55: Quality Function Deployment Approach to Target Prioritization.

Filled circles indicate a strong relationship, open circles depict a moderate relationship, triangles indicate a weak relationship, and empty boxes indicate that there is no relationship between the strategic objective and the target set. The percentages at the bottom of the figure represent the relative importance of each target set with respect to the importance rankings for each strategic objective and are calculated according to Equation 8:

$$(\text{TargetSetImportance})_j = \prod_{i=1}^{n} (\text{StrategicObjective})_i \, (\text{Interrelationship})_{ij} \tag{8}$$

The multiple attribute decision making technique described in Section B.1 mathematically ranks concepts based on the data that describes a concept and the subjective importance of the various data categories. While the subjective importance for each target set can be defined from the output of the QFD in Figure 55, ranking of the nearly 450 targets in the simulation requires a means of specifying the "threat value" in each target set. A representation of the overall priority of a given target, T, is the threat value for target T in target set j multiplied by the relative importance of that target set with respect to the strategic objectives as defined in the QFD exercise as shown below using an Overall Evaluation Criterion (OEC):

$$(\text{Priority})_T = \sum_{j=1}^{12} (\text{ThreatValue})_{Tj} \, (\text{TargetSetImportance})_j \tag{9}$$

Note that the threat value for each target, T, is defined using the pseudo-signature method detailed in Section 5.3.2.3. Using the QFD matrix and the overall evaluation criterion specified above, each target can be given an overall priority where the highest priority indicates the most important target on the battle manager's list.

This ranking scheme is depicted in Figure 56 for a Phase I-like campaign where the primary objectives are the establishment of air supremacy and the degradation of major NBC facilities. In this case, the highest priority targets are nuclear research facilities, IADS sector operations centers, communications relays, high-profile leadership targets, and mobile Scud missile launchers. Using alternative settings for the strategic objectives, the same algorithm can be applied to a Phase IV-like operation where the battlefield is prepared for ground forces. In this case (Figure 57), the primary targets are the Republican Guard headquarters, Petroleum/Oil/Lubricant (POL) storage facilities, and refineries. Army units, roads, bridges, and storage facilities become a priority in this scenario. It is also important to note how the pie chart depicting the relative importance of target sets changes between the two disparate scenarios. While this graphical depiction shows how the QFD and OEC can be used to prioritize targets, this algorithm was converted to FLAMES code and placed inside the GITBattleManagerRank function. As shown in the activity diagram in Figure 50, the rank function is executed every time a new target is nominated by the battle manager. When the target list is updated, the battle manager searches through the list of available subordinates to see if any asset can successfully engage the highest priority target. A technique for matching platform/weapon combinations to high priority targets using intelligent agents is discussed in the next section.

	Objective	Importance	Leadership	Electricity	Telecom/C3	IADS	NBC	TBM Launcher	Air Force	Navy	Army	Oil	Roads/Bridges	Mil. Storage
National Policy Objectives (CINCCENT Mission Statement)	Neutralize Iraqi National Command Authority	10	⊙	○	⊙	△	○	○	△	△	○			
	Eject Iraqi Armed Forces from Kuwait	0			○					○	⊙	○		⊙
	Destroy the Republican Guard	0	○	△							⊙			⊙
	Destroy Iraq's Ballistic Missile and NBC Capability	5	○	△			⊙	⊙	△	△	△			△
	Assist in the Restoration of the Legitimate Gov't of Kuwait	0	△								△			
Other Strategic Objectives	Obtain and Maintain Air Supremacy	8	△		○	⊙			⊙					△
	Cut Supply Lines to the KTO	0	○	○	○						⊙	⊙	⊙	⊙
	Destroy Saddam's Capability to Threaten Neighbors	2.5	⊙	○	⊙	⊙	⊙	⊙	⊙	△	⊙	○	○	○
			16%	5%	16%	13%	12%	12%	13%	2%	7%	1%	1%	2%

Target Name	Priority
Nuke- Tuwaitha Nuclear Research Center	1.00
IOC Talil (Fighters)	1.00
IOC Taqqadum (Fighters)	1.00
SOC H3	1.00
SOC Kirkuk	1.00
SOC Taji	1.00
SOC Talil	1.00
Baghdad- Communications Relay Station	0.96
Baghdad- New Presidential Palace	0.95
Baghdad- Presidential Palace (Command Bunker)	0.95
Baghdad- Presidential Palace (Command Center)	0.95
Baghdad- Presidential Residence	0.95
IOC Taji (Fighters)	0.91
Baghdad- Ministry of Propaganda	0.85
Baghdad- Ministry of Defense	0.84
Baghdad- Presidential Palace North	0.84
Scud 01	0.83
Scud 02	0.83
Scud 03	0.83
Scud 04	0.83
Scud 05	0.83

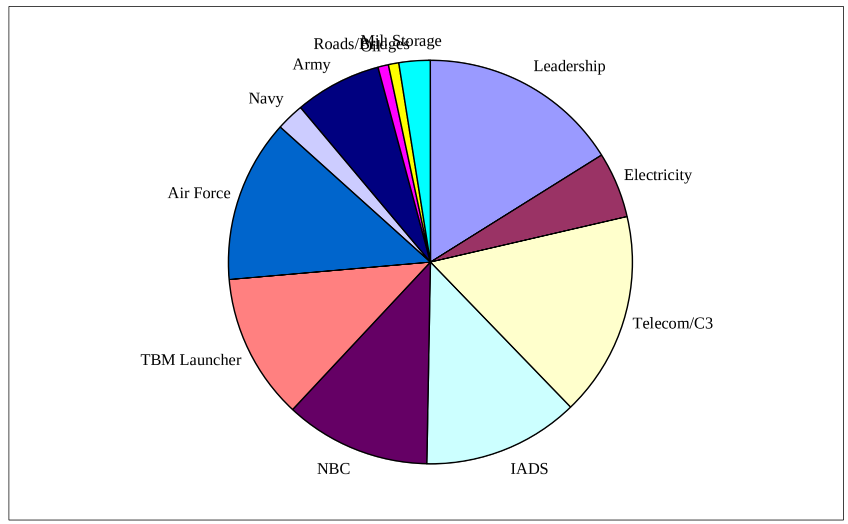


Figure 56: Target Prioritization Interactive Interface for Phase I: Strategic Air Campaign.

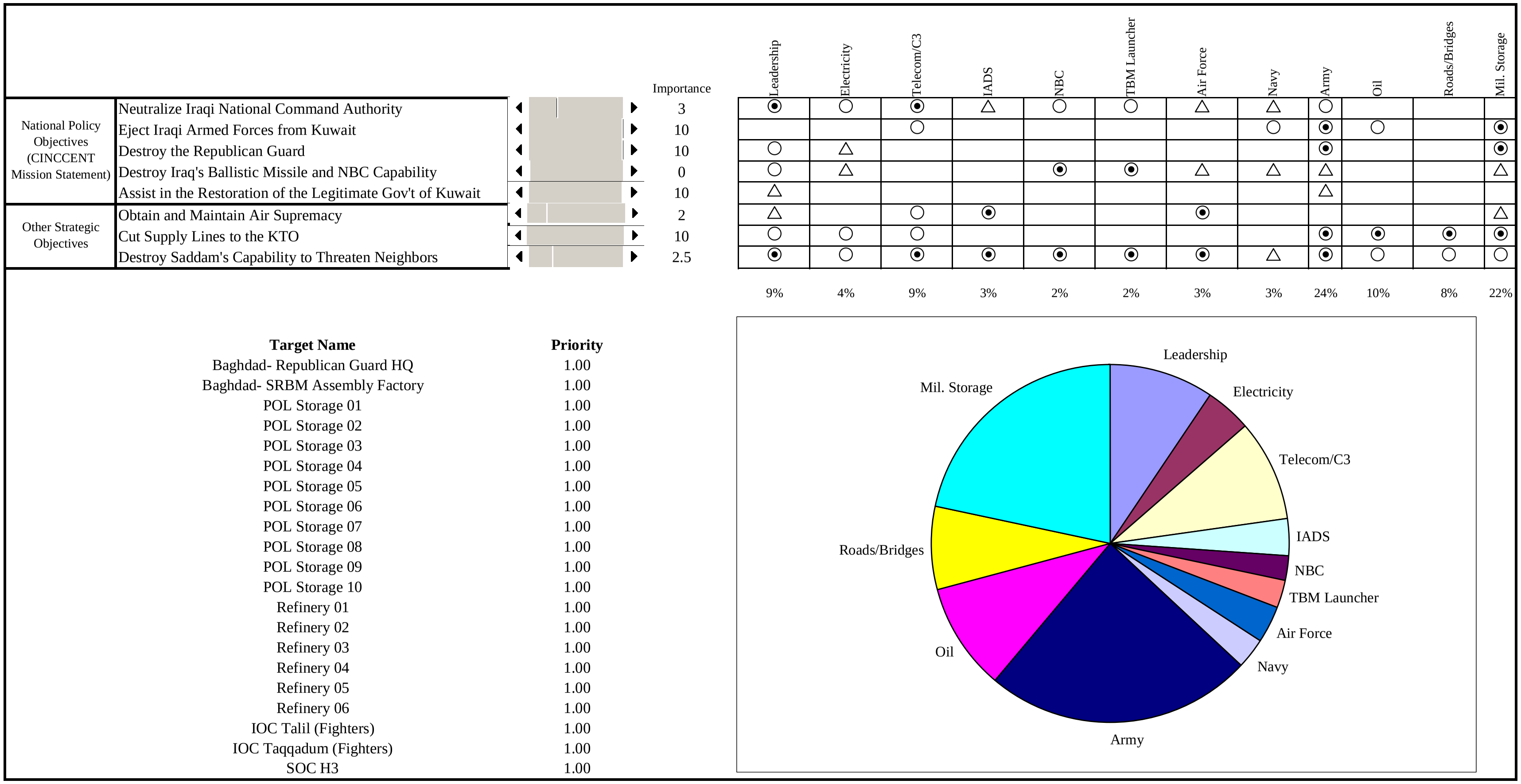


Figure 57: Target Prioritization Interactive Interface for Phase IV: Support Ground Campaign to Liberate Kuwait.

5.5 Step 5: Develop and Train Intelligent Battle Manager

"Strategy without tactics is the slowest route to victory. Tactics without strategy is the noise before defeat."

-Sun Tzu

Targeting is the "process of selecting and prioritizing targets and matching the appropriate response to them, taking account of operational requirements and capabilities," the first part of which is currently handled by the battle manager [468]. Section 5.4.1 described a technique for arranging targets in order of importance based on the target set(s) to which they belong. This gives the battle manager the first level of intelligence: it knows *what* it wants to do, but not *how* to best do it.

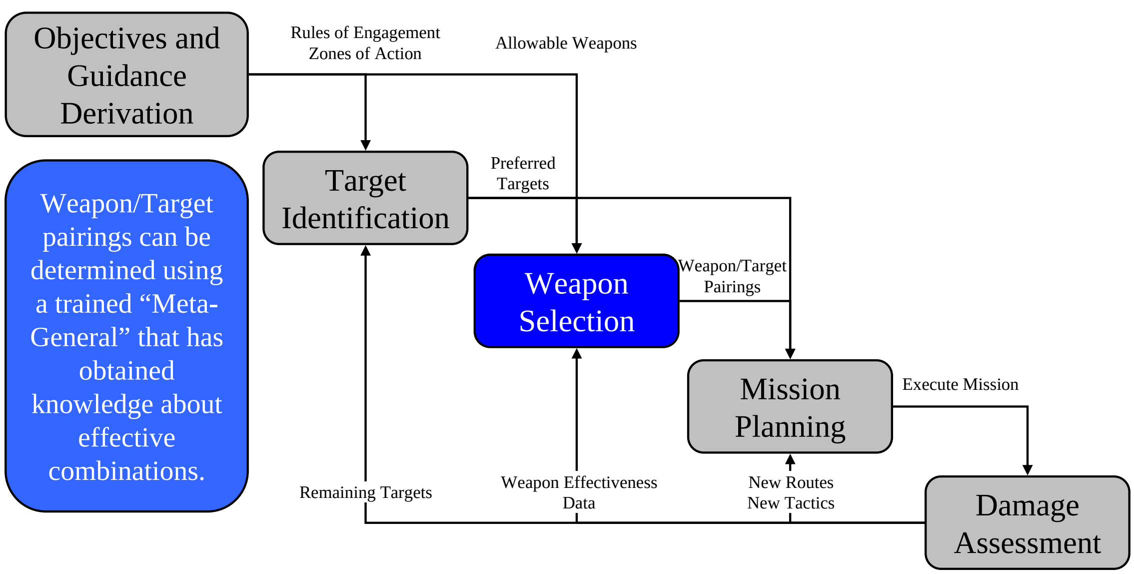


Figure 58: The Role of Weaponeering in the Target Cycle (Adapted from Reference [429]).

The next function in the Air Force Target Cycle (see Figure 53) is weaponeering, which is "the process of determining the quantity of a specific type of lethal or nonlethal weapons required to achieve a specific level of damage to a given target" [468]. These two functions work together to match weapons to targets to achieve a desired effect. The battle manager's domain is limited to targeting and weaponeering, and after these functions are completed, the output is the ATO which essentially lists the weapon/target pairs which have been selected to achieve the warfighter's desired effects for the mission at hand.

A cumbersome method for emulating the real-world behavior of weaponeers would be to create a table that maps target types to weapon types and allow the battle manager to use this table lookup to map weapons to targets at runtime. Such a table would be full of bias and reflect the knowledge of experienced weaponeers, which is unavailable for this task. If distinct rules could be defined to emulate the behavior of real-world weaponeers, a decision tree (see Section A.2.4) could be created that represents the allowable decisions. Unfortunately, a hardcoded decision tree would not be adaptable for new weapons and technologies. Only existing options on the tree would be available to the battle manager. Additionally, the search time to fathom the decision tree at each decision point for each potential target would be excessive. An alternative is to define the physical and functional properties of each target and allow the battle manager to *learn* which weapon/target pairings are effective.

The lack of computational resources and experienced weaponeers drives the decision to utilize agent-based modeling and machine learning to discover valid asset/weapon/target pairings depending on the state of the scenario and the assets and technologies available. The proposed approach uses two modes of operation, training and analysis, to create an intelligent "Meta-General" that has experience in a number of battlefield scenarios. The general description of this approach is given in Section B.2.1. To avoid the shortcomings of a decision tree, a neural network surrogate model is used to encapsulate the "intelligence" of the Meta-General and enable rapid execution at runtime.

5.5.1 Training the Intelligent Battle Manager

To train the battle manager, a "simple" scenario was defined that retains the properties of the IADS developed for the testbed scenario, but uses only a single target and a single strike aircraft (bomber). While all the properties of the target except its geographic location are held constant, the properties of the strike aircraft can be varied to represent a wide range of potential platforms and munitions. The ranges of the system and subsystem level metrics for the battle manager training exercise are shown in Table 5.

The technology parameters shown in Table 5 enable exploration of a range of platforms

Table 5: Ranges for the Design of Experiments to Train the Battle Manager.

Variable	**Low**	**High**
Max Speed (Mach)	0.72	4
Cruise Altitude (m, ft)	3,048 (10,000)	15,240 (50,000)
GTOW (kg, lbs)	15,876 (35,000)	544,311 (1,200,000)
Empty Wt Ratio	0.4	0.55
Payload Wt (kg, lbs)	907 (2,000)	36,287 (80,000)
Thrust/Weight	0.35	1.5
Wing Loading (lb/ft^2)	20	150
Drag Coefficient	0.01	0.09
Max C_L	1.5	3
RCS (m^2)	0.01	10
TSFC (lb$_m$/lb$_f$-hr)	0.3	0.8
Munition Range (km, nm)	18.52 (10)	2,222 (1,200)
Munition Speed (Mach)	0.72	6
SAM Density (%)	0	100

(in the air domain) detailed in Sections 2.4.2 and 2.4.3 as well as a wide range of munitions from the existing arsenal and notional future concepts. For example, platform speeds up to Mach 4 and munition speeds up to Mach 6 are encompassed by these ranges. Additionally, by varying the starting and ending geographic coordinates of both platforms[19], the threat context and platform range constraints can be captured. Coordinates were varied to represent a range of about 2,800 km (1,500 nautical miles), defined by Shlapak, et. al as a reasonable range to conduct high-tempo operations on the Arabian Peninsula [373]. Finally, a missing element is the density of SAM sites in the region. If a platform flies over a dense area of SAMs, the battle manager will learn that direct paths between those two coordinates are undesirable; however, if the SAM sites have already been destroyed, this path becomes acceptable. A routine was written in FLAMES to randomly kill a user-defined percentage of SAM sites around the conflict region. Since the same random seed was used for each simulation, the SAM sites are always destroyed in the same order. The simulation setup for the battle manager training exercise is shown in Figure 59.

The purpose of training the Meta-General is to expose it to many possible situations

[19]The design of experiments table defines two square regions that encompass Iraq and the areas south of Iraq in which most platforms are located. Defining these ranges as square regions allows the target to be placed in either Syria or Iran; however, the battle manager onlys allow engagements within the borders of Iraq.

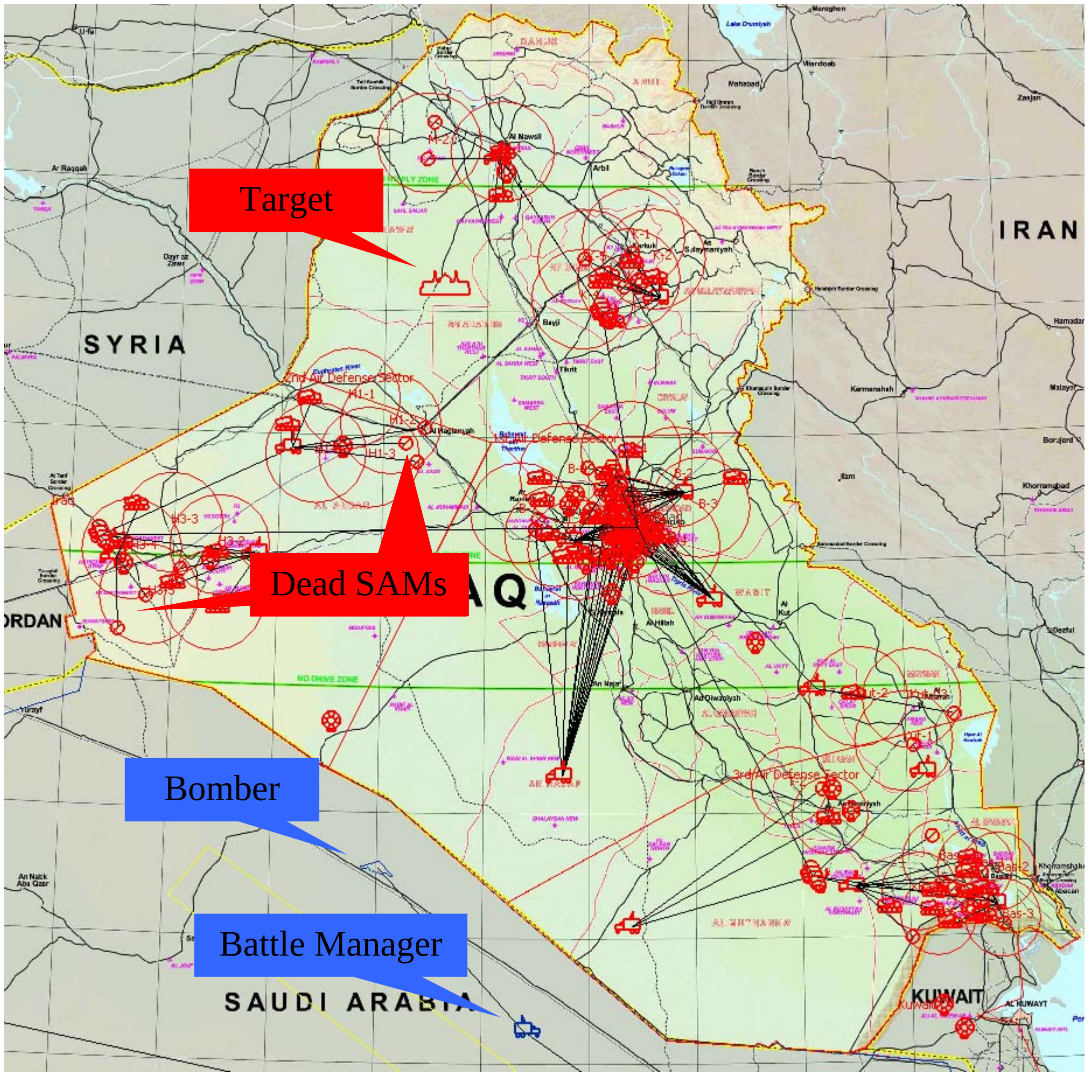


Figure 59: Scenario for the Battle Manager Training Exercise (Background Image from Reference [315]).

to build a database of what weapons and platforms are effective. As an initial test, 4,000 random points were executed across the range of input parameters defined in Table 5. All parameters except the target location were uniformly distributed throughout the range of input parameters in Table 5. The target location was biased so that more than 50% of the training targets were within the boundaries of Baghdad, which is consistent with the actual distribution of targets and defenses in Iraq during the 1991 Persian Gulf War.

The DOE was executed using two Pentium IV computers for a total of 11 CPU-hours. Of the 4,000 cases run, in 800 instances the target of interest was placed outside the borders of Iraq. These cases confound the training of the battle manager because the platform either gives up or attempts to attack a SAM site. In the actual simulation, targets outside Iraq would not be prosecuted by the battle manager. These failed cases are an artifact of creating a geometric square that must encompass an irregularly shaped region.

While the regression on these cases was underway, an additional 6,000 random cases were run using the same computer setup over a period of approximately 16 CPU-hours. Of the 10,000 total cases, 8,922 were usable.

5.5.2 Summary of Meta-General Training Data

A summary of the 8,922 valid cases used in the training of the battle manager is shown in Figure 60. This type of plot is called the *multivariate profiler* and is useful for determining the relationship between parameters for systems-of-systems. The term “multivariate” refers to any process that considers multiple variables simultaneously. In contrast to simplified one or two dimensional analyses, the multivariate profiler provides the designer with a view of all variable interactions simultaneously. Along the diagonal, the variable names are listed. The convention used in this work places more general variables (MoEs) in the upper left corner and specific technology or design parameters in the lower left corner. The top level measures of effectiveness for targets killed and platforms lost are shown in the upper left hand corner. Along the diagonal, platform design variables, munition design variables, and geographic parameters are also indicated. Each of the boxes above the diagonal shows the relationship between the two variables that comprise the intersection while all other variables are also simultaneously varied over the ranges shown. This depiction is analogous to viewing the *total derivative* of the design space across the range of all parameters in the system-of-systems hierarchy. Designers use this plot to gain insight into the behavior of the design space and should use a combination of colors, symbols, density plots, and constraint lines to understand the character of the space.

In Figure 60, blue points represent “successful engagements” where the blue platform survived and the red target was killed. Red points show the opposite case, an “unsuccessful

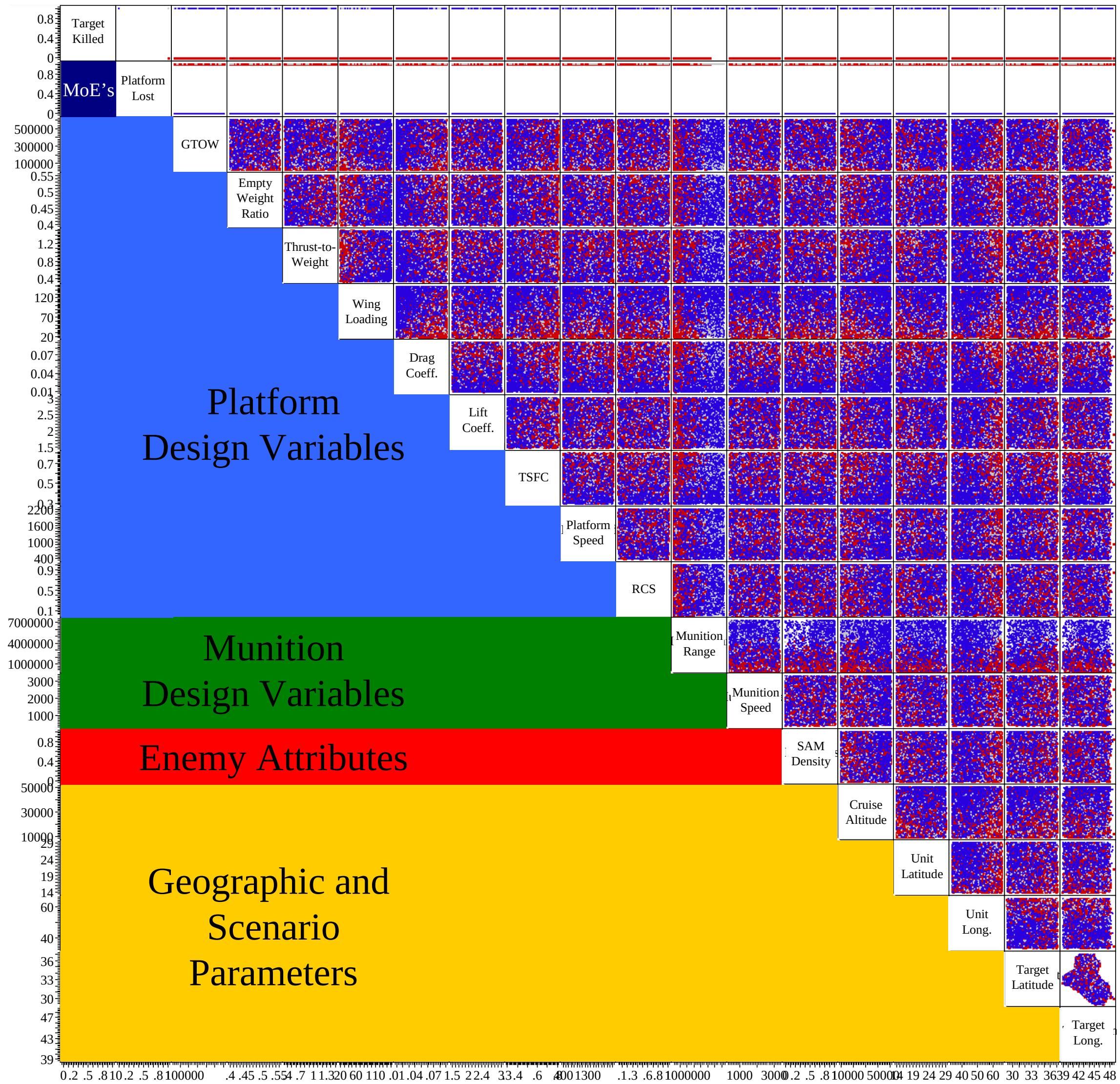


Figure 60: Multivariate Plot Illustrating the Results of the Battle Manager Training.

engagement" where the friendly platform died and the hostile target survived. The remainder of the cases are indicative of neutral engagements, where both entities died. In this case, it is not clear *when* the friendly platform died, or if the death resulted from missile fire or fuel depletion. From the coloration in Figure 60, several interesting trends are visible. First, it appears that blue points dominate the region of long munition range. Second, as shown in Figure 61, successful engagements are more prevalent at high wing loading and low thrust-to-weight ratio. A higher thrust-to-weight ratio (at cruise) is not favored because higher weight leads to higher thrust. For a given cruise TSFC, greater thrust corresponds to a greater fuel burn. For a given gross takeoff weight and a defined fuel fraction, a platform with greater thrust-to-weight ratio will therefore have reduced range or run out of fuel before completing its mission. The plot in Figure 61 supports the fact that bombers tend to have high wing loading and low thrust-to-weight ratios [278]. Higher thrust-to-weight ratios would tend to be valued by the battle manager if the asset under test engaged in air-to-air combat with extremely capable adversaries.

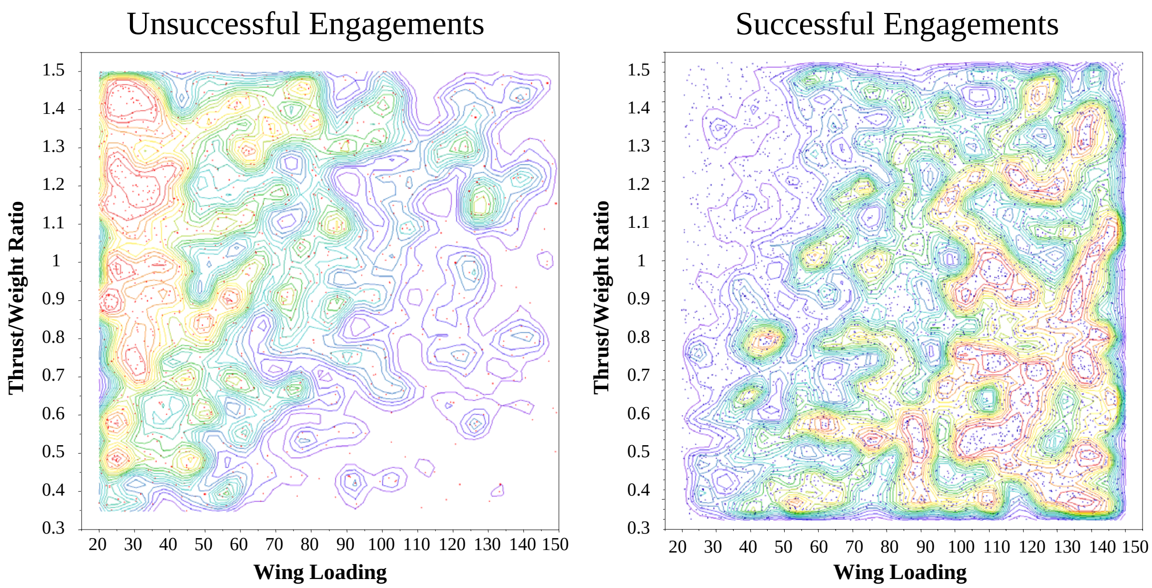


Figure 61: Successful Engagements Tend to Favor High Wing Loading and Low Thrust/Weight Ratio (High Density Shown in Red).

Other trends from the battle manager training are summarized in Table 6 for successful, neutral, and unsuccessful engagements. The "difference" column is indicative of the variation between successful and unsuccessful engagements.

While parameters like gross takeoff weight, empty weight ratio, lift coefficient, radar cross section, and munition speed do not seem to contribute significantly to bomber effectiveness in this scenario, design parameters such as wing loading, thrust-to-weight ratio, drag coefficient, TSFC, cruise altitude, and munition range contribute significantly to the success of the engagement. This is because the contribution of the second set of parameters is so much larger than the first set, it appears they have little or no influence. This result is related to the fact that all degrees of freedom are simultaneously varying.

Table 6: Analysis of Mean for Successful, Neutral, and Unsuccessful Engagements for Battle Manager Training.

Parameter	Successful	Neutral	Unsuccessful	Difference
Gross Takeoff Weight (kg)	146,006	141,181	136,083	-6.8%
Empty Weight Ratio	0.4723	0.4793	0.4812	1.9%
Thrust-to-Weight Ratio	0.8931	0.9743	1.0055	12.6%
Wing Loading (lb/ft^2)	91.91	72.54	64.45	-29.9%
Drag Coefficient	0.046	0.059	0.061	32.4%
Lift Coefficient	2.25	2.24	2.23	-0.7%
TSFC (lb$_m$/lb$_f$-hr)	0.5266	0.5858	0.6110	16.0%
Platform Speed (Mach)	1.92	2.14	2.11	9.9%
RCS (m^2)	0.5038	0.5244	0.5091	1.0%
Munition Range (km)	752	829	389	-48.3%
Munition Speed (Mach)	2.84	2.81	2.80	1.3%
Unit Altitude (m)	9,731	8,019	7,545	-22.5%
Data Points	6,361	1,154	1,392	

Munition range has the single greatest impact on mission success since the success criteria were defined in terms of the ability to kill a target without losing the aggressor platform. Scaling parameters, drag coefficient, and TSFC all relate to the platform's ability to complete its mission without running out of fuel. According to the training results, a higher altitude is favored, although it is not clear whether this effect is due to the decreased density (and hence increased range) or a decrease in SAM effectiveness at higher altitudes. Further investigation into the cause of platform losses is required to definitively answer this question. Finally, the most curious result is that munition speed has no impact on the success of the mission. This is because the target being used for this engagement is a fixed target. It is expected that munition speed has a first-order impact on mission success against moving targets; however, additional cognition and physics models must be created to support the prosecution of moving targets.

5.5.3 Results of the Meta-General Training Experiment

If the research questions in Section 3.2 can be subdivided into Grand Strategic, Strategic, Operational, and Tactical levels, the research questions that arise within this section can best be termed "procedural." For example, in medicine a patient may be identified as being ill, their illness may be classified as cancer, identified as a certain type of cancer, located within the body, and a treatment regiment that includes radiation therapy may be prescribed. All of these categorical determinations lend themselves well to downselection by morphology. On the other hand, procedural questions such as "how much radiation to deliver?" are more quantitative and are not easily expressed using a matrix of alternatives. Many such procedural research questions arise in the training of the Meta-General and are answered by an exploratory process of testing and verifying hypotheses. For example:

- What kind of surrogate model is most appropriate?
- What is the optimum topology of the neural network?
- Is the validation set error, test set error, or the overall predictive ability of the equation a better method for evaluating the overall "goodness" of a neural network?
- What level of predictive error is "acceptable" for the trained Meta-General?
- How should computer resources be allocated between the amount of time to spend on training, and the number of iterations to perform during Meta-General training?
- How do different types of DOEs compare in terms of their ability to generate a valid decision model?
- How many cases are needed to develop a valid fit for the neural network?

Each of the above procedural research questions must be completely answered to not only validate the proposed approach to using a Meta-General to replace human decision makers (answers the operational question "Can it be done?") but also to specify the procedures that must be performed at the lowest level to use the Meta-General approach effectively (answers the tactical question "How can it *best* be done?").

The first situation examined was the development of a Meta-General for fixed targets. Originally, it was believed that a single neural network would be able to handle targets of all characteristics; however, results from the Meta-General training (see Section 5.5.2) identified several input parameters including munition speed that have little effect on the success of an engagement. For this reason, a second training algorithm for moving targets was developed after additional cognition models for moving target prosecution were created. The results of the moving target training exercise are detailed in Section 5.5.9.

5.5.4 Neural Networks Outperform Polynomial Response Surface Equations by a Large Margin

Section C.7 identifies polynomial response surface equations and neural networks as two effective types of surrogate models. The non-linear nature of the military system-of-systems problem does not lend itself well to approximation using polynomial surrogates. The neural network technique is an enabler to model discontinuities endemic to this class of problems.

5.5.4.1 The Optimum Topology Can Be Determined Using BRAINN

Topology in network theory refers to the structure or layout of a network in terms of nodes and connections. To train a backpropagation neural network, the number of input, output, and hidden nodes must be explicitly specified. The BRAINN tool developed by Johnson and Schutte iteratively tests multiple network topologies to identify the optimum network configuration for a given set of data. In the Meta-General training procedure, a given set of data can be analyzed using this tool by specifying the range of hidden nodes to try. The number of input nodes is a function of the number of input parameters (in this case 17) and the number of output nodes is defined by the number of responses (in this case two). To maximize the effectiveness of the method while minimizing the computational resources expended, at least two passes are used for each set of data. In the first case, a "coarse pass" is used where the training time and number of iterations is lowered and the number of hidden nodes to examine is variable. After this pass identifies the optimum topology, the training process is repeated using the optimum number of hidden nodes and the coefficients of the equation are identified.

5.5.5 Four Measures of Error are Available

There are four primary types of error analysis used for the neural network equation. The first error measure is the error in the training set. This error measure should be extremely low, because it defines how well the neural network was able to match the data used to create the neural network and is analogous to the model fit error defined by Kirby and Barros [55]. The second error measure used is the validation set. The validation set is used to assess the ability of the network to generally match behavior throughout the design space. Johnson and Schutte note that "by using the validation set to determine the 'optimal' network the program is slightly distorting the validation set as a measure of the generalization error. It is no longer an independent test of the network's generalizability [225]." In contrast, the test set is a set of random data that is neither explicitly or implicitly used for neural network training. This data set is usually a much smaller percentage of the overall data set and is analogous to the model representation error identified by Kirby and Barros [55]. The final measure of error is the total error (also called predictive error) of the equation which examines the overall ability of the neural network equation to predict data values in the training, validation, and test sets as well as any other data points that are added that were not included in the data used to train the neural network. This measure is calculated in JMP® by assessing how many times is the neural network correct in comparison to the total number of cases in the data set.

While the test set is the most independent measure of the generalizability of the neural network, this data is usually a small portion of the overall data set (typically 3-5%). For this reason, the test set error must be compared against the total error, which is a measure of how well the neural network can predict values across the entire design space. While low test set error for any size neural network is a good property, maintaining low total error in the presence of large data sets may be equally valuable.

5.5.6 An Acceptable Error Threshold is 5%

These results implicitly define the next procedural question: "what is an acceptable level of error?" Given the myriad of interacting factors that are involved in the simulation of a military system-of-systems, an ability to forecast weapon/platform/target pairings within 5% accuracy is likely sufficient for the purposes intended. The resulting data shown in Table 11 indicate that the Meta-General training approach proposed in this dissertation is extremely accurate at forecasting such pairings.

5.5.6.1 Computational Resources Should be Allocated to Training Time

Another procedural question addresses how computational resources should be apportioned between two user-defined settings in the BRAINN graphical interface: training time and number of iterations. The training time is a period of time over which the training algorithm attempts to maximize the R^2 value of the validation set, while the number of iterations is the number of times this process repeats for different initial conditions. The total time required for the training process is given by Equation 10.

$$TotalTime_{hr} = \left[\frac{Nodes_{High} - Nodes_{Low} + 1}{Increment}\right]\left[\frac{(Iterations)\,(TrainingTime_s)}{3600}\right] \qquad (10)$$

Where the $Nodes_{Low}$ and $Nodes_{High}$ define the starting and ending number of hidden nodes to examine. Generally, the total time is constrained to a period of approximately 12 hours (overnight) and the remaining parameters are set accordingly.

A comparison was performed between two training cases for a 10,000 case data set run through the FLAMES simulation and regressed using the BRAINN tool. In the first case, the training time was set higher than the number of iterations. The opposite was true for the second case. The fit for the equation "platforms lost" was used for comparison. The resulting error distributions between the two cases are shown in Table 7.

Table 7: Comparison Between Two Training Cases to Determine Apportionment Between Training Time and Iterations.

Parameter	**Case 1 Training Time Focus**	**Case 2 Iterations Focus**
Validation Data (%)	25	25
Test Data (%)	3	3
Training Time (s)	**300**	**120**
Hidden Nodes	10	10
Iterations at Each	**200**	**400**
Training % Correct	97.5583	97.3872
Validation % Correct	93.1034	92.6108
Test % Correct	93.6567	92.9104
Total % Correct	96.256	96.110
Number of Cases	10,000	10,000

The predictive error for Case 1 was 3.74% (374/10,000) while the error for Case 2 was 3.87% (387/10,000). While Case 1 outperforms Case 2 across the board, the training times used were generally long enough to obtain excellent fits for the data used. The results shown in Table 7 confirm the hypotheses that Meta-General training is more effective when the training time is larger and the number of iterations is smaller. This gives the Meta-General more time to examine the topology as opposed to giving it more chances to start over with new initial guesses. Through a comparative experiment, it was determined that a longer training time for a lower number of iterations minimizes the error of the resulting neural network.

5.5.7 Comparison of DOE Types for Meta-General Training

Since the field of machine learning and artificial intelligence has yet to generalize the type of DOE that is best suited for training an intelligent agent, a comparison between two options for DOEs was conducted. Johnson recommends a space-filling DOE supplemented with a central composite design (CCD) for the generation of neural network surrogate models [225]. A DOE consisting of purely random cases is generally easier to create than a space-filling DOE, and both are easier to create than a Latin Hypercube. The first test is to compare the correlation between independent variables in a random DOE consisting of 10,000 cases and a sphere-packed lattice with the same number of cases. The random DOE was generated using the `RandomUniform()` function in JMP® 6.0, while the lattice was created using the MATLAB® Model Based Calibration toolbox [274]. Each DOE was created in approximately ten seconds; however, creating a space-filling DOE in JMP® 6.0 takes considerably longer. Both DOEs were examined using the multivariate analysis feature in JMP® and the magnitude of the correlation coefficients between independent variables were calculated and are shown in Figure 62. Red shading indicates that the magnitude of the correlation is greater than the opposing method, while green shading indicates that the magnitude of the correlation is lower. From the coloration in Figure 62, it is obvious that a space-filling scheme has much lower correlation than a purely random DOE.

To further analyze Johnson's recommendation, a latin hypercube was created for 10,000 cases. The addition of the constraint of uniformity increases the generation time from 10 seconds to 72 minutes. For this reason, any benefit gained from using a latin hypercube design of experiments must be weighed against the computational burden that increases as more cases are required and more design variables are added.

Comparison of a Random Design of Experiments to a Space-Filling Design of Experiments

	GTOW	EWR	TWR	WS	CD	CL	TSFC	MaxSpeed	RCS	Munition Range	Munition Speed	Num Sams	Unit Lat	Unit Lon	Unit Alt	Target Lat	Target Lon
GTOW	1	0.0023	0.0015	0.0164	0.0144	0.0102	0.002	0.0045	0.0023	0.0005	0.0035	0.013	0.0114	0.0174	0.0046	0.0092	0.0061
EWR	0.0023	1	0.0029	0.005	0.0136	0.0172	0.0052	0.0143	0.0178	0.0057	0.0038	0.0109	0.0051	0.0053	0.0017	0.0178	0.0068
TWR	0.0015	0.0029	1	0.0021	0.0111	0.0016	0.0092	0.0054	0.001	0.0055	0.0063	0.0018	0.0085	0.0112	0.0113	0.0093	0.0173
WS	0.0164	0.005	0.0021	1	0.0046	0.0075	0.0103	0.0165	0.0103	0.0056	0.0032	0.0064	0.0017	0.0204	0.0075	0.007	0.0122
CD	0.0144	0.0136	0.0111	0.0046	1	0.0072	0.0177	0.0153	0.0058	0.0095	0.0069	0.0038	0.0017	0.0042	0.0049	0.0065	0.0011
CL	0.0102	0.0172	0.0016	0.0075	0.0072	1	0.0092	0.0105	0.0089	0.0059	0.0049	0.0046	0.005	0.0142	0.0057	0.0135	0.0125
TSFC	0.002	0.0052	0.0092	0.0103	0.0177	0.0092	1	0.0023	0.0006	0.0029	0.0038	0.0027	0.0067	0.0094	0.0028	0.0046	0.021
MaxSpeed	0.0045	0.0143	0.0054	0.0165	0.0153	0.0105	0.0023	1	0.0189	0.002	0.0019	0.0107	0.0003	0.0172	0.0133	0.0089	0.0115
RCS	0.0023	0.0178	0.001	0.0103	0.0058	0.0089	0.0006	0.0189	1	0.004	0.031	0.0257	0.0076	0.01	0.0003	0.0071	0.0014
Munition Range	0.0005	0.0057	0.0055	0.0056	0.0095	0.0059	0.0029	0.002	0.004	1	0.0072	0.2507	0.0103	0.0062	0.0071	0.085	0.0737
Munition Speed	0.0035	0.0038	0.0063	0.0032	0.0069	0.0049	0.0038	0.0019	0.031	0.0072	1	0.031	0.0091	0.0071	0.0102	0.0091	0.0059
Num Sams	0.013	0.0109	0.0018	0.0064	0.0038	0.0046	0.0027	0.0107	0.0257	0.2507	0.031	1	0.0043	0.0144	0.0034	0.0528	0.0563
Unit Lat	0.0114	0.0051	0.0085	0.0017	0.0017	0.005	0.0067	0.0003	0.0076	0.0103	0.0091	0.0043	1	0.0221	0.0007	0.0101	0.0031
Unit Lon	0.0174	0.0053	0.0112	0.0204	0.0042	0.0142	0.0094	0.0172	0.01	0.0062	0.0071	0.0144	0.0221	1	0.0072	0.0044	0.0047
Unit Alt	0.0046	0.0017	0.0113	0.0075	0.0049	0.0057	0.0028	0.0133	0.0003	0.0071	0.0102	0.0034	0.0007	0.0072	1	0.0028	0.0032
Target Lat	0.0092	0.0178	0.0093	0.007	0.0065	0.0135	0.0046	0.0089	0.0071	0.085	0.0091	0.0528	0.0101	0.0044	0.0028	1	0.3231
Target Lon	0.0061	0.0068	0.0173	0.0122	0.0011	0.0125	0.021	0.0115	0.0014	0.0737	0.0059	0.0563	0.0031	0.0047	0.0032	0.3231	1

Comparison of a Space-Filling (Sphere Packing Scheme) Design of Experiments to a Random Design of Experiments

	GTOW	EWR	TWR	WS	CD	CL	TSFC	MaxSpeed	RCS	Munition Range	Munition Speed	Num Sams	Unit Lat	Unit Lon	Unit Alt	Target Lat	Target Lon
GTOW	1	0.0001	0.0019	0.0005	0.0014	0.0044	0.0028	0.0009	0.0003	0.0018	0.0007	0.0005	0.0076	0.0006	0.0004	0.0013	0.0017
EWR	0.0001	1	0.0015	0.0001	0.0008	0.0031	0.0046	0.0009	0.0012	0.0012	0.0003	0.0006	0.0013	0.0017	0.0012	0.0006	0.0003
TWR	0.0019	0.0015	1	0.0033	0.006	0.001	0.0002	0.0019	0.0013	0.0011	0.0002	0.0001	0.0036	0.0018	0.0011	0.0009	0.0008
WS	0.0005	0.0001	0.0033	1	0.0004	0.0004	0.0014	0.0008	0.0014	0	0.0026	0.0009	0.0014	0.0001	0.0038	0.0005	0.0014
CD	0.0014	0.0008	0.006	0.0004	1	0.0008	0.0007	0.0002	0.001	0.0005	0.001	0.0003	0.0042	0.0018	0.0006	0	0.0034
CL	0.0044	0.0031	0.001	0.0004	0.0008	1	0.0003	0.001	0.0009	0.0005	0.0001	0.0022	0.001	0	0.0005	0.0015	0.0029
TSFC	0.0028	0.0046	0.0002	0.0014	0.0007	0.0003	1	0.0012	0.0016	0.0013	0.0026	0.0007	0.001	0.0034	0.0017	0.0003	0.0001
MaxSpeed	0.0009	0.0009	0.0019	0.0008	0.0002	0.001	0.0012	1	0.0035	0.0011	0.0018	0.0005	0.0014	0.0013	0.0003	0.0051	0.0054
RCS	0.0003	0.0012	0.0013	0.0014	0.001	0.0009	0.0016	0.0035	1	0.0021	0.0024	0.002	0.0002	0.0002	0.0016	0.0028	0.0019
Munition Range	0.0018	0.0012	0.0011	0	0.0005	0.0005	0.0013	0.0011	0.0021	1	0.0006	0.0014	0.0029	0.0006	0.0002	0.0022	0.0007
Munition Speed	0.0007	0.0003	0.0002	0.0026	0.001	0.0001	0.0026	0.0018	0.0024	0.0006	1	0.0015	0.0037	0.0011	0.0012	0.0001	0.0001
Num Sams	0.0005	0.0006	0.0001	0.0009	0.0003	0.0022	0.0007	0.0005	0.002	0.0014	0.0015	1	0.0004	0.0001	0.0004	0.0009	0.0019
Unit Lat	0.0076	0.0013	0.0036	0.0014	0.0042	0.001	0.001	0.0014	0.0002	0.0029	0.0037	0.0004	1	0.0003	0.0008	0.0001	0.0001
Unit Lon	0.0006	0.0017	0.0018	0.0001	0.0018	0	0.0034	0.0013	0.0002	0.0006	0.0011	0.0001	0.0003	1	0.0008	0.0001	0.0005
Unit Alt	0.0004	0.0012	0.0011	0.0038	0.0006	0.0005	0.0017	0.0003	0.0016	0.0002	0.0012	0.0004	0.0008	0.0008	1	0.0026	0.0002
Target Lat	0.0013	0.0006	0.0009	0.0005	0	0.0015	0.0003	0.0051	0.0028	0.0022	0.0001	0.0009	0.0001	0.0001	0.0026	1	0.0001
Target Lon	0.0017	0.0003	0.0008	0.0014	0.0034	0.0029	0.0001	0.0054	0.0019	0.0007	0.0001	0.0019	0.0001	0.0005	0.0002	0.0001	1

Figure 62: Comparison of the Magnitude of the Correlation Between Independent Variables for Two Types of DOEs.

Comparison of Latin Hypercube to the Random Design of Experiments

	GTOW	EWR	TWR	WS	CD	CL	TSFC	MaxSpeed	RCS	Munition Range	Munition Speed	Num Sams	Unit Lat	Unit Lon	Unit Alt	Target Lat	Target Lon
GTOW	1	0.0008	0.002	0.0218	0.0291	0.0044	0.0136	0.0179	0.0025	0.0152	0.0129	0.0028	0.0108	0.0088	0.0046	0.0009	0.0094
EWR	0.0008	1	0.0031	0.0148	0.0007	0.001	0.0168	0.0011	0.0031	0.0091	0.0226	0.0166	0.0196	0.0031	0.0023	0.0005	0.0155
TWR	0.002	0.0031	1	0.0092	0.015	0.0191	0.0186	0.002	0.004	0.0075	0.0081	0.0114	0.0022	0.0078	0.0034	0.0158	0.0172
WS	0.0218	0.0148	0.0092	1	0.0069	0.0182	0.0163	0.006	0.0079	0	0.007	0.018	0.0014	0.0078	0.006	0.0046	0.0034
CD	0.0291	0.0007	0.015	0.0069	1	0.0095	0.0037	0.0022	0.003	0.0047	0.0028	0.0046	0.002	0.0188	0.0053	0.008	0.0004
CL	0.0044	0.001	0.0191	0.0182	0.0095	1	0.0136	0.003	0.0028	0.0043	0.0123	0.0127	0.0273	0.003	0.0063	0.0213	0.0102
TSFC	0.0136	0.0168	0.0186	0.0163	0.0037	0.0136	1	0.0055	0.0183	0.0099	0.0071	0.0101	0.0088	0.0017	0.0138	0.0045	0.0052
MaxSpeed	0.0179	0.0011	0.002	0.006	0.0022	0.003	0.0055	1	0.0082	0.0077	0.011	0.0047	0.0027	0.011	0.0161	0.0147	0.0049
RCS	0.0025	0.0031	0.004	0.0079	0.003	0.0028	0.0183	0.0082	1	0.0075	0.007	0.0031	0.003	0.0048	0.0113	0.0234	0.0078
Munition Range	0.0152	0.0091	0.0075	0	0.0047	0.0043	0.0099	0.0077	0.0075	1	0.0047	0.0046	0.0133	0.0196	0.0087	0.0171	0.0004
Munition Speed	0.0129	0.0226	0.0081	0.007	0.0028	0.0123	0.0071	0.011	0.007	0.0047	1	0.0117	0.0068	0.0014	0.0159	0.0045	0.017
Num Sams	0.0028	0.0166	0.0114	0.018	0.0046	0.0127	0.0101	0.0047	0.0031	0.0046	0.0117	1	0.0099	0.0224	0.0118	0.0129	0.0013
Unit Lat	0.0108	0.0196	0.0022	0.0014	0.002	0.0273	0.0088	0.0027	0.003	0.0133	0.0068	0.0099	1	0.0004	0.0045	0.0106	0.0063
Unit Lon	0.0088	0.0031	0.0078	0.0078	0.0188	0.003	0.0017	0.011	0.0048	0.0196	0.0014	0.0224	0.0004	1	0.0051	0.0093	0.0015
Unit Alt	0.0046	0.0023	0.0034	0.006	0.0053	0.0063	0.0138	0.0161	0.0113	0.0087	0.0159	0.0118	0.0045	0.0051	1	0.0044	0.0202
Target Lat	0.0009	0.0005	0.0158	0.0046	0.008	0.0213	0.0045	0.0147	0.0234	0.0171	0.0045	0.0129	0.0106	0.0093	0.0044	1	0.0087
Target Lon	0.0094	0.0155	0.0172	0.0034	0.0004	0.0102	0.0052	0.0049	0.0078	0.0004	0.017	0.0013	0.0063	0.0015	0.0202	0.0087	1

Comparison of a Latin Hypercube to a Space-Filling (Sphere Packing Scheme) Design of Experiments

	GTOW	EWR	TWR	WS	CD	CL	TSFC	MaxSpeed	RCS	Munition Range	Munition Speed	Num Sams	Unit Lat	Unit Lon	Unit Alt	Target Lat	Target Lon
GTOW	1	0.0008	0.002	0.0218	0.0291	0.0044	0.0136	0.0179	0.0025	0.0152	0.0129	0.0028	0.0108	0.0088	0.0046	0.0009	0.0094
EWR	0.0008	1	0.0031	0.0148	0.0007	0.001	0.0168	0.0011	0.0031	0.0091	0.0226	0.0166	0.0196	0.0031	0.0023	0.0005	0.0155
TWR	0.002	0.0031	1	0.0092	0.015	0.0191	0.0186	0.002	0.004	0.0075	0.0081	0.0114	0.0022	0.0078	0.0034	0.0158	0.0172
WS	0.0218	0.0148	0.0092	1	0.0069	0.0182	0.0163	0.006	0.0079	0	0.007	0.018	0.0014	0.0078	0.006	0.0046	0.0034
CD	0.0291	0.0007	0.015	0.0069	1	0.0095	0.0037	0.0022	0.003	0.0047	0.0028	0.0046	0.002	0.0188	0.0053	0.008	0.0004
CL	0.0044	0.001	0.0191	0.0182	0.0095	1	0.0136	0.003	0.0028	0.0043	0.0123	0.0127	0.0273	0.003	0.0063	0.0213	0.0102
TSFC	0.0136	0.0168	0.0186	0.0163	0.0037	0.0136	1	0.0055	0.0183	0.0099	0.0071	0.0101	0.0088	0.0017	0.0138	0.0045	0.0052
MaxSpeed	0.0179	0.0011	0.002	0.006	0.0022	0.003	0.0055	1	0.0082	0.0077	0.011	0.0047	0.0027	0.011	0.0161	0.0147	0.0049
RCS	0.0025	0.0031	0.004	0.0079	0.003	0.0028	0.0183	0.0082	1	0.0075	0.007	0.0031	0.003	0.0048	0.0113	0.0234	0.0078
Munition Range	0.0152	0.0091	0.0075	0	0.0047	0.0043	0.0099	0.0077	0.0075	1	0.0047	0.0046	0.0133	0.0196	0.0087	0.0171	0.0004
Munition Speed	0.0129	0.0226	0.0081	0.007	0.0028	0.0123	0.0071	0.011	0.007	0.0047	1	0.0117	0.0068	0.0014	0.0159	0.0045	0.017
Num Sams	0.0028	0.0166	0.0114	0.018	0.0046	0.0127	0.0101	0.0047	0.0031	0.0046	0.0117	1	0.0099	0.0224	0.0118	0.0129	0.0013
Unit Lat	0.0108	0.0196	0.0022	0.0014	0.002	0.0273	0.0088	0.0027	0.003	0.0133	0.0068	0.0099	1	0.0004	0.0045	0.0106	0.0063
Unit Lon	0.0088	0.0031	0.0078	0.0078	0.0188	0.003	0.0017	0.011	0.0048	0.0196	0.0014	0.0224	0.0004	1	0.0051	0.0093	0.0015
Unit Alt	0.0046	0.0023	0.0034	0.006	0.0053	0.0063	0.0138	0.0161	0.0113	0.0087	0.0159	0.0118	0.0045	0.0051	1	0.0044	0.0202
Target Lat	0.0009	0.0005	0.0158	0.0046	0.008	0.0213	0.0045	0.0147	0.0234	0.0171	0.0045	0.0129	0.0106	0.0093	0.0044	1	0.0087
Target Lon	0.0094	0.0155	0.0172	0.0034	0.0004	0.0102	0.0052	0.0049	0.0078	0.0004	0.017	0.0013	0.0063	0.0015	0.0202	0.0087	1

Figure 63: Comparison of the Magnitude of the Correlation of a Latin Hypercube DOE to the Random and Space-Filling DOEs.

While Figure 62 compared the correlation coefficients of the random DOE and the space-filling DOE to each other, Figure 63 compares the 10,000 case latin hypercube correlation to both the random design of experiments (top) and the space-filling DOE (bottom). When compared to the random DOE, a universal judgement cannot be easily made as to which is better: it appears to be a wash. On the other hand, a space-filling DOE is universally better than the latin hypercube. Almost all the correlation coefficients between independent variables are higher for a latin hypercube created using the same tool as the space-filling DOE. Surprisingly, the additional time expenditure required to generate a latin hypercube for this problem results in a less effective design than a sphere-packing scheme that can be created almost instantly.

While this experiment demonstrated that a space-filling design tends to have the lowest independent variable correlation, the next step is to ascertain whether the quality of low independent variable correlation matters when training the intelligent battle manager. For this experiment, two types of DOEs are compared. The first, a random DOE consisting of 10,000 points was created and run through the simulation. Of these, only 8,922 were usable.

Additionally, a 15,000 case space-filling DOE was created using the model-based calibration toolbox and supplemented with a 512 case central composite design created using JMP®. One key problem with this setup is that by definition, the CCD tends to place points at the extremes of the design space. In the case of the geographic parameters, target coordinates outside Iraq are excluded. As a result, the CCD cases were nearly all excluded. Of the 15,000 cases in the space-filling DOE, only 8,132 were inside the borders of Iraq and were considered valid. This number of cases is close to the number of cases run for the random DOE so that the subsequently generated neural network equations can be compared against the same basis. The following sections describe the process for creating a neural network equation for the two DOE types. In general, this process follows two steps: first, the topology of the neural network equation must be determined. This involves running a shorter training time for an number of network configurations to identify the ideal number of hidden nodes. Once the topology has been identified, a second pass is conducted that optimizes the coefficients based on the selected topology by using longer training times and more iterations for only a single network configuration. In some cases, the error is so low after the first pass that a second pass is not necessary.

5.5.7.1 *Random DOE Exploratory Phase (Case 1): Determining the Topology of the Neural Network*

A neural network equation was used to approximate the two primary outputs of the training: whether a blue platform was lost and whether the red target was killed. The regression was performed using the BRAINN interface developed by Johnson and Schutte [225]. The maximum and minimum values for the hidden nodes were defined using past experience and an observation that topologies with greater than 10-15 nodes may overfit the data. The fit statistics for this experiment are shown in Table 8.

Table 8: Results from Battle Manager Neural Network Training, Random DOE Exploratory Phase.

	Response	
Parameter	**Platforms Lost**	**Targets Killed**
Validation Data (%)	25	25
Test Data (%)	1	1
Training Time (s)	120	120
Hidden Nodes (low)	6	10
Hidden Nodes (high)	11	16
Iterations at Each	100	100
Training % Correct	95.9385	98.5217
Validation % Correct	93.4169	95.4322
Test % Correct	N/A	N/A
Optimal Nodes	10	10
Number of Cases	8922	8922

The total percent error for the "platforms lost" equation was 4.62% (413/8922) and the percent error for the "targets killed" equation was 2.21% (197/8922). An analysis of the distribution of the error indicated that there was no bias to any particular region of the design space and the error still exhibited random scattering. The main outcome of Case 1 is the identification of the optimum number of hidden nodes as 10 for each of the two responses.

5.5.7.2 *Random DOE Optimization Phase (Case 2): Optimizing the Neural Network Using the Identified Topology*

To improve the fit of the neural network equations, the second iteration of training was run using the same data with a longer training time and more iterations for a topology that uses 10 hidden nodes. The results of this training run are shown in Table 9.

Table 9: Results from Battle Manager Neural Network Training, Random DOE Optimization Phase.

	Response	
Parameter	**Platforms Lost**	**Targets Killed**
Validation Data (%)	25	25
Test Data (%)	3	3
Training Time (s)	300	300
Hidden Nodes	10	10
Iterations at Each	200	200
Training % Correct	97.5583	98.5692
Validation % Correct	93.1034	95.2978
Test % Correct	93.6567	93.6567
Number of Cases	8922	8922

The total percent error for the "platforms lost" equation was 3.74% (334/8922) and the percent error for the "targets killed" equation was 2.36% (211/8922). Although the additional training time at the optimal number of hidden nodes did not commensurately improve the accuracy of the equation, the total error of both equations was less than 5%.

5.5.7.3 *Space-Filling DOE Exploratory Phase (Case 1): Determining the Topology of the Neural Network*

As with the random DOE, an initial coarse training pass was used to identify the optimum topology for the space-filling DOE. The resulting goodness of fit statistics for the resulting neural network equations are shown in Table 10.

The total percent error for the "platforms lost" equation was 0.92% (75/8132) and the total percent error for the "targets killed" equation was 0.38% (31/8132). Because the percent error is so low for the equations identified in this round, further training of the neural network to optimize the coefficients is not necessary.

Table 10: Results from Battle Manager Neural Network Training, Sphere-Packing Scheme Exploratory/Optimization Phase.

	Response	
Parameter	**Platforms Lost**	**Targets Killed**
Validation Data (%)	25	25
Test Data (%)	3	3
Training Time (s)	120	120
Hidden Nodes (low)	5	5
Hidden Nodes (high)	10	10
Iterations at Each	100	100
Training % Correct	99.4193	99.7438
Validation % Correct	98.0817	99.213
Test % Correct	99.1803	100.00
Optimal Nodes	9	10
Number of Cases	8132	8132

5.5.7.4 Summary: A Space-Filling DOE is Most Effective for Meta-General Training

The initial procedural equation that defined this experiment was "how do different types of DOEs compare in terms of their ability to generate a valid decision model?"

When comparing the optimal results for a random DOE (Table 9) and a space-filling DOE (Table 10), a space-filling DOE universally outperforms a random DOE. While both DOEs cover the same range of input variables, the space-filling DOE has lower independent variable correlation and all measures of error are unequivocally lower. Furthermore, the space-filling DOE had slightly fewer cases and required less training to converge to a better result. Generally, the space-filling DOE should have validation and test set points that are randomly generated, as opposed to actual points in the space-filling lattice. Although the lattice points were randomly selected for error analysis, a combination of a lattice DOE for training and a random DOE for validation and test could be used to further improve the model fit. For this reason, another training pass was performed using the full 17,054 cases to maximize the ability of the neural network to cover the design space while minimizing predictive error. The goodness of fit statistics from the optimal training pass for each type of DOE are summarized in Table 11. As this table shows, the space-filling DOE always outperforms the random DOE.

Table 11: Summary of the Comparison Between Random and Space-Filling DOEs for Meta-General Training.

	Random DOE		**Space-Filling DOE**	
Parameter	**Platforms Lost**	**Targets Killed**	**Platforms Lost**	**Targets Killed**
Training % Correct	97.5583	98.5692	99.4193	99.7438
Validation % Correct	93.1034	95.2978	98.0817	99.213
Test % Correct	93.6567	93.6567	99.1803	100.00
Overall % Correct	96.256	97.6351	99.08	99.619
Number of Cases	8922	8922	8132	8132

5.5.8 Developing the Final Form of the Neural Network Equation

For a final training pass, the two previously used data sets were combined to produce a single DOE that combines the properties of a space-filling DOE with a random DOE. 4,000 of the 8,132 random DOE points were used for the validation set (23%) and 500 points were used for the test set (3%). The results of the exploratory DOE are shown in Table 12.

Table 12: Results from Battle Manager Neural Network Training, Combined DOE Exploratory Phase.

	Response	
Parameter	**Platforms Lost**	**Targets Killed**
Validation Data (%)	25	25
Test Data (%)	3	3
Training Time (s)	120	120
Hidden Nodes (low)	5	5
Hidden Nodes (high)	10	10
Iterations at Each	100	100
Training % Correct	97.5336	99.2757
Validation % Correct	95.7571	92.8
Test % Correct	95.1172	93.8
Optimal Nodes	8	9
Number of Cases	17,054	17,054

From these results, the optimum topology was identified as eight and nine hidden nodes for the "platforms lost" and "targets killed" responses respectively. The results of the final optimization pass using the combined data set are summarized in Table 13.

The total percent error for the "platforms lost" equation was 3.46% (590/17054) and the total percent error for the "targets killed" equation was 2.56% (437/17054). At first

Table 13: Results from Battle Manager Neural Network Training, Combined DOE Optimization Phase.

	Response	
Parameter	**Platforms Lost**	**Targets Killed**
Validation Data (%)	23	23
Test Data (%)	3	3
Training Time (s)	300	300
Hidden Nodes	8	9
Iterations at Each	200	200
Training % Correct	98.3603	99.2836
Validation % Correct	91.575	92.425
Test % Correct	90.60	91.20
Total % Correct	96.54	97.44
Number of Cases	17,054	17,054

glance, the predictive capability of the combined data set appears inferior to the neural network devised using only a space-filling design (Table 10). Despite the fact that all error measures increased, does the greater coverage of the design space improve the *overall* fit of the neural network equation?

When the neural network created in Section 5.5.7.3 using the pure space-filling DOE was applied to the full 17,054 case data set, the total error increased from 0.92% to 12.8% for the "platforms lost" response and from 0.38% to 6.9% for the "targets killed" response. The error distribution was disproportionately allocated to the random DOE points that were not used to train the previous neural network. This means that the neural network created using solely the space-filling DOE is not as accurate with off-DOE points as previously expected.

From the series of experiments in this section, it is evident that although the space-filling neural network is the most accurate, the final form of the neural network equation to be used in the simulation activity is the variant that combines data from both previous trials to create a single data set of 17,054 runs.

The lessons learned from the previous tests are applied to the training of the Meta-General for moving targets: instead of using a random DOE, a space-filling DOE of at least 20,000 points is executed. When training the Meta-General, the amount of "experience" provided is a key measure of the performance of the neural network: the training set should include as many cases as are reasonable given the computational constraints of the analysis. The results of all training passes are summarized in Figures 64 and 65. One key observation from these figures is that the percent correct is generally higher for the "targets killed" response than the "platforms lost" response. This is partially due to the fact that the settings used for the DOE resulted in a "kill" approximately 85% of the time and a "loss" only about 32% of the time: there is simply more data available to understand the cause of a kill than there is to determine potential causes for a loss. Also, although the neural network created using only the space-filling DOE appears to have the highest accuracy, as described in Section 5.5.8, it is not as accurate at predicting off-DOE points for the full 17,054 case set.

When taking into account a balance of design space coverage and predictive error, **the best overall equation that most thoroughly approximated the space of potential engagements with minimum predictive error was the neural network that resulted from the 17,054 case combined DOE**, although all training passes resulted in acceptable equations with less than 5% total error. The goodness of fit statistics for all of the tested equations are shown in Figures 64 and 65 below.

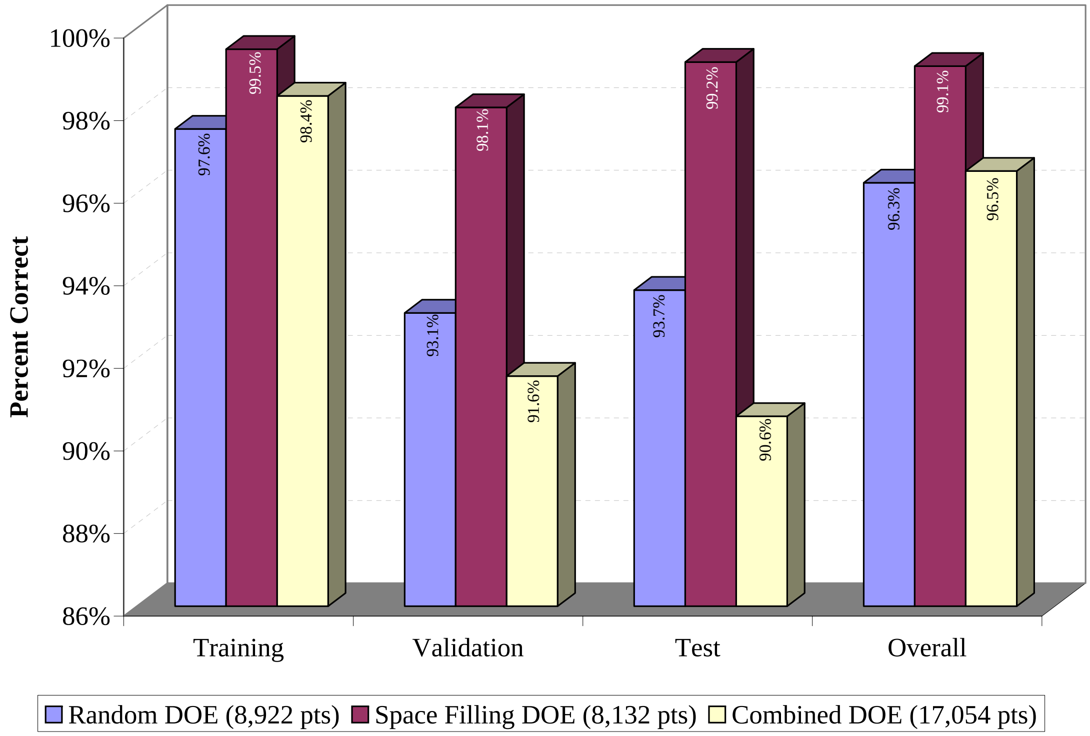


Figure 64: Error Comparison for Three Different Training Cases- Platforms Lost.

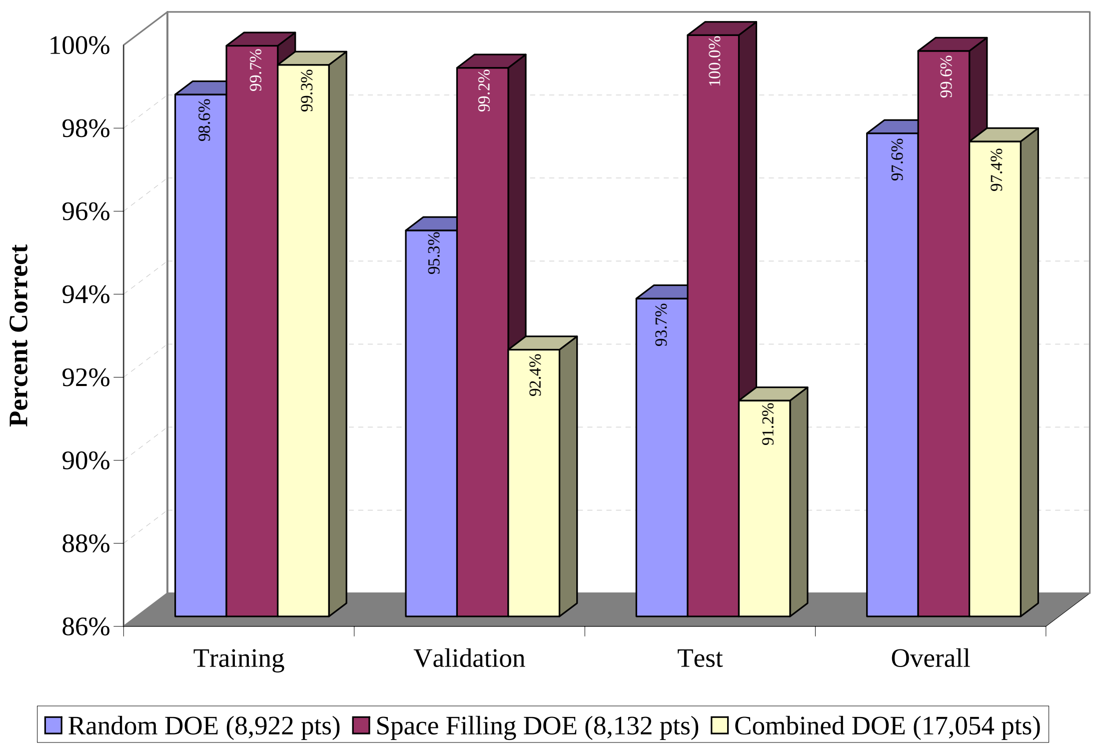


Figure 65: Error Comparison for Three Different Training Cases- Targets Killed.

5.5.9 Meta-General Training for Moving Targets

The first experiment with Meta-General training in Section 5.5.3 revealed a shortcoming of the initial experimental setup: certain performance inputs produced no realizable benefit on the system although they were originally expected to do so. Most notable of these was the weapon speed. After analyzing the data, it was determined that this resulted from the focus of the training on fixed targets. The battle manager had no logic to deal with moving targets. While the trained battle manager described in Section 5.5.3 is valid for the HDBT, decapitation strike, and conventional strike missions, it is not appropriate for the TCT attack mission. This sparked the development of a modified cognition model that could prosecute moving targets.

The cognition model for the GITBomber was designed to obtain target information from an initial sensor track and plot a course to that target. When the bomber was within weapon range, the weapon would be released and begin its fly-out to the target; however, the GITBomber had no on-board sensor to *track* the target after it received the initial coordinates. As a result, if the target relocated while the bomber was enroute to the target, the weapon would miss its intended target unless the lethal radius of the munition was extraordinarily large.

This realization led to the development of a more sophisticated cognition model based on the FCFighter example cognition model that was originally ruled out in favor of the FCAIN example model as described in Section 5.3.2.2. An activity diagram for the GITFCFighter cognition model and supporting modules is shown in Figure 66. While the GITBattleManager looks for targets using the same processes used for fixed targets, if a target is identified as a moving or movable target, it is assigned to a platform using the GITTimeCriticalTargetAssign function as opposed to the ground controller module developed for fixed targets. The fighter cognition model was modified to receive ground targets from the Time Critical Target Controller, and establishes a vector to the last known coordinates of the target. Once this waypoint is within the range of the onboard sensor, the GITFCFighter activates its sensor and begins to look for the target.

Meanwhile, a TBM launcher follows the activity diagram shown in Figure 67. The green

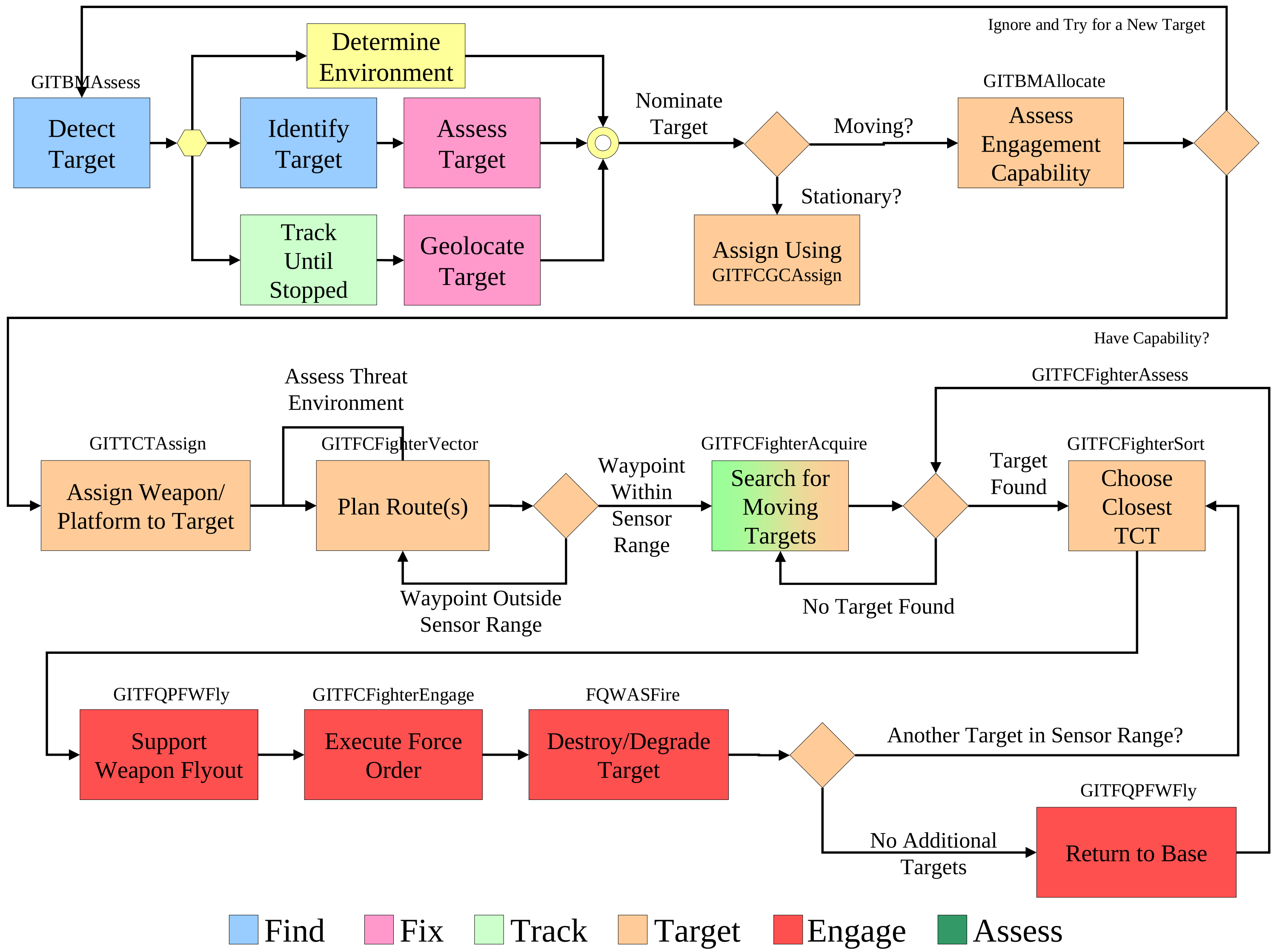


Figure 66: Activity Diagram for Time Critical Strike (Adapted from [134]).

boxes indicate states where the target is visible, while the red boxes enumerate states in which the target is hidden. The TBM must be visible to fire and move, although this does not guarantee that the target can be detected during these phases.

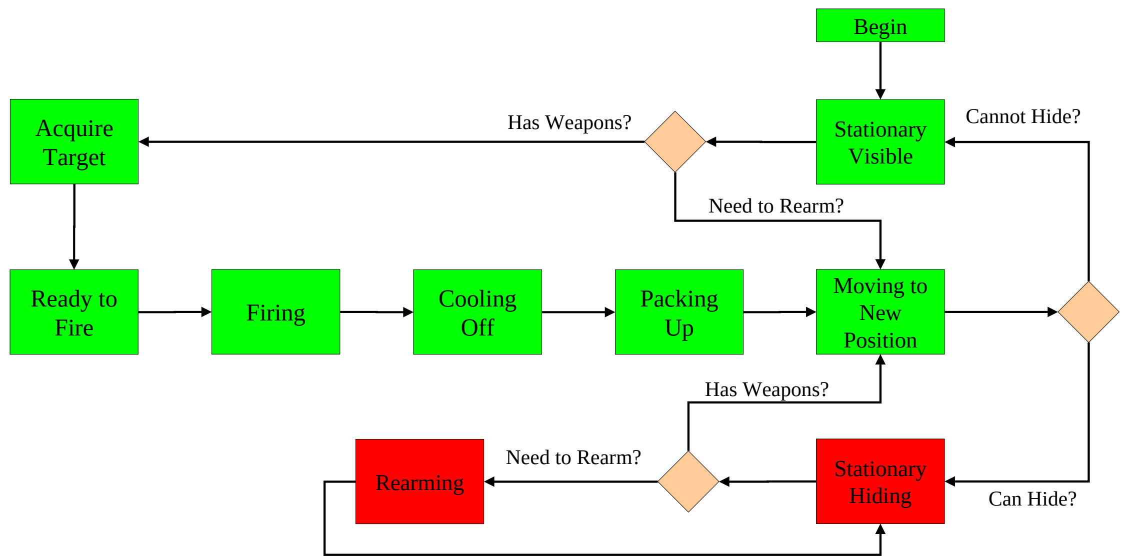


Figure 67: Activity Diagram for a Theater Ballistic Missile Launcher.

If the TBM is in a visible state and the blue fighter is attempting to acquire the target and is within sensor range of the target, the GITFCFighter attempts to choose the closest TCT to its current position. The fighter then vectors towards the target and release a homing weapon using the onboard weapon system. After releasing the weapon, the GITFCFighter will then attempt to acquire another target in its sensor range. If another TCT is found, the engagement process continues as long as the aggressor finds additional targets. When no targets are found near the vector coordinates, the platform returns to base. While returning to base, the fighter executes the ASSESS method in continuous mode, which attempts to search for any additional TCTs along its flight path as it returns to base. Additionally, after the fighter begins to return to base, it is flagged as available for tasking by the Time Critical Target Controller. While in theater, the fighter is much closer to potential targets than a fighter at a distant base. Provided that a fighter has both fuel and munitions available, it is highly likely to be tasked to intercept additional TCTs while in the theater if they are detected by the Time Critical Target Controller.

The scenario used to train the battle manager against moving targets is shown in Figure 68. Based on the lessons learned from the training of the Meta-General for fixed targets, a space-filling DOE was created using the ranges of inputs shown in Table 14. The original DOE was populated with 45,000 cases; however, after cases outside Iraq were excluded, only 25,822 were executed through FLAMES in approximately 50 CPU hours. Of these, 24,271 (94%) resulted in valid engagements. Preprocessing the DOE to remove extraneous cases saved approximately 37 CPU hours. Also, in addition to adding several parameters related to the moving target properties, the blue bomber was allowed to start inside Iraq as well as anywhere on the Arabian peninsula in contrast to the fixed target training where most start locations were south of Iraq.

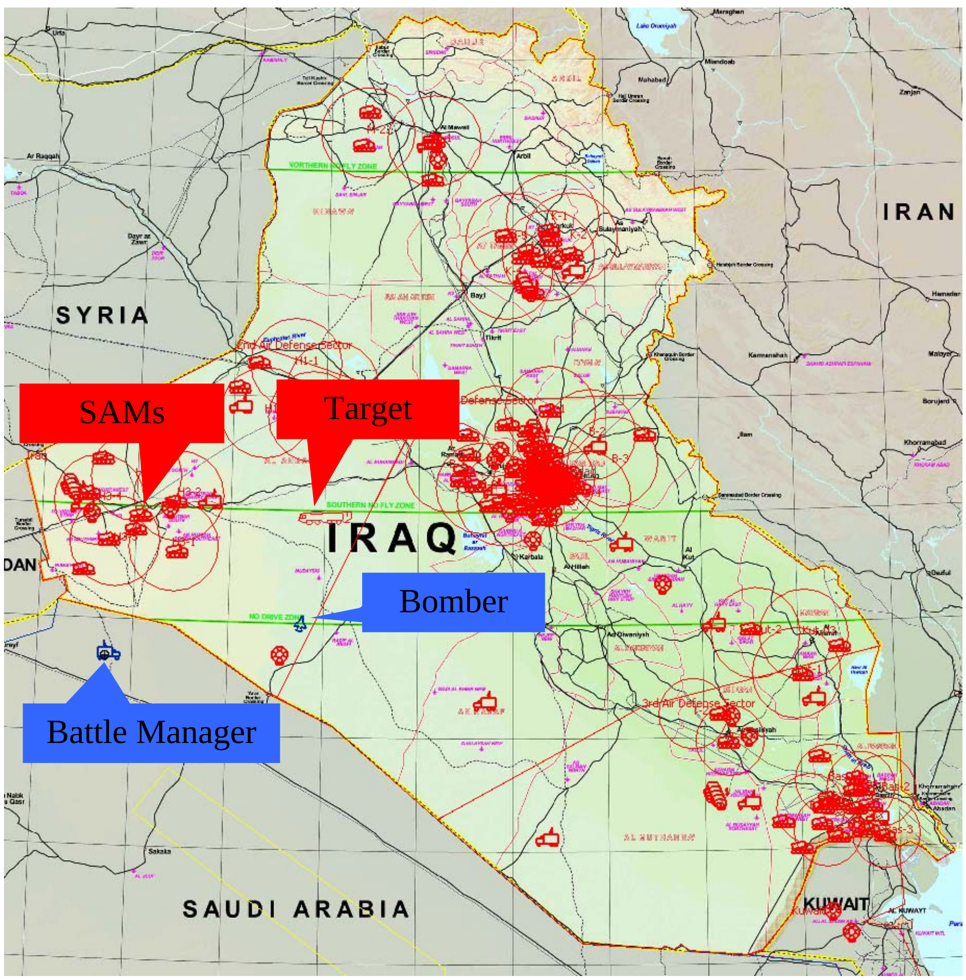


Figure 68: Scenario for the Battle Manager Training Exercise (Background Image from Reference [315]).

Table 14: Ranges for the Design of Experiments to Train the Battle Manager for Moving Targets.

Variable	**Low**	**High**
Max Speed (Mach)	0.72	4
Cruise Altitude (m, ft)	3,048 (10,000)	15,240 (50,000)
GTOW (kg/lbs)	15,876 (35,000)	544,311 (1,200,000)
Empty Wt Ratio	0.4	0.55
Thrust/Weight	0.35	1.5
Wing Loading (lb/ft^2)	20	150
Drag Coefficient	0.01	0.09
Max C_L	1.5	3
RCS (m^2)	0.01	1
TSFC (lb_m/lb_f-hr)	0.3	0.8
Munition Range (km, nm)	1.85 (1)	2,408 (1,300)
Munition Speed (Mach)	0.72	6
Sensor Range (km, nm)	1.85 (1)	2,222 (1,200)
Target Speed (m/s, mph)	0	22.35 (50)
Target RCS (m^2)	0	10
Target Heading (deg)	0	360
Target Movement Time (hrs)	0	1
SAM Density (%)	0	100

The data from the moving target training experiment are shown in Figures 69 and 70 where blue points indicate situations where the target is killed without losing a platform (11,489/24,271), and red points indicate situations where the opposite is true (1,392/24,271). The remaining points are situations where either both died (204/24,271) or where no engagement took place because the platform was unable to locate the target (11,186/24,271).

The ANOVA technique can be applied to the output data to examine the impact on the two responses, resulting in the Pareto chart shown in Figure 71. The green shaded area corresponds to the 80% threshold of importance. While munition speed still has little direct impact on the responses, the *dwell time* parameter is calculated as:

$$DwellTime = \frac{MunitionRange}{MunitionSpeed} \tag{11}$$

And the munition is automatically destroyed if its flight time exceeds the dwell time. Therefore, munition speed is implicit in the dwell time parameter.

While the Pareto chart is useful in ascertaining the dominant factors, it is also important to develop an understanding of how design parameters impact different solution regimes.

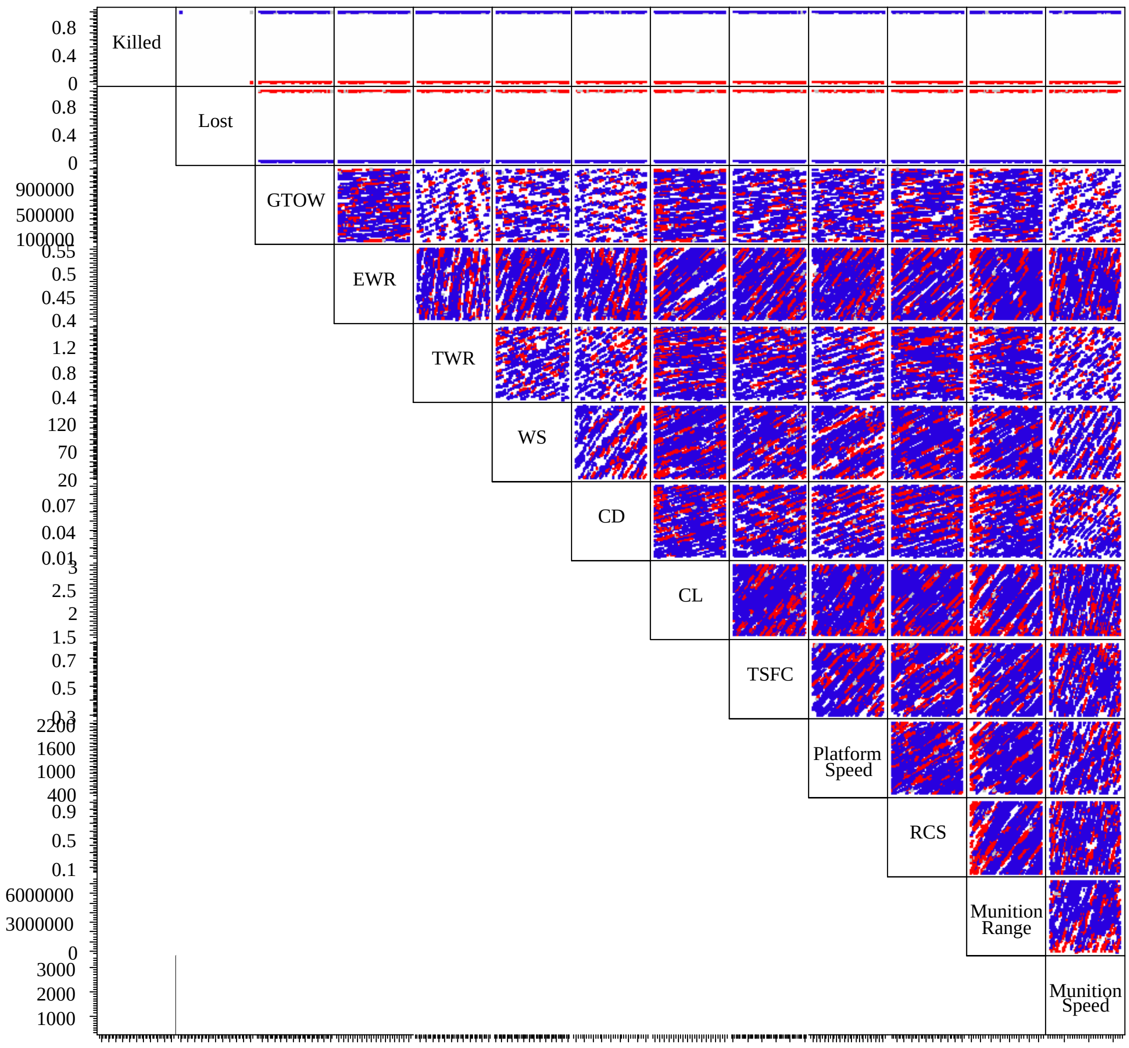


Figure 69: Multivariate Profiler for the Moving Target Training Experiment (1 of 2).

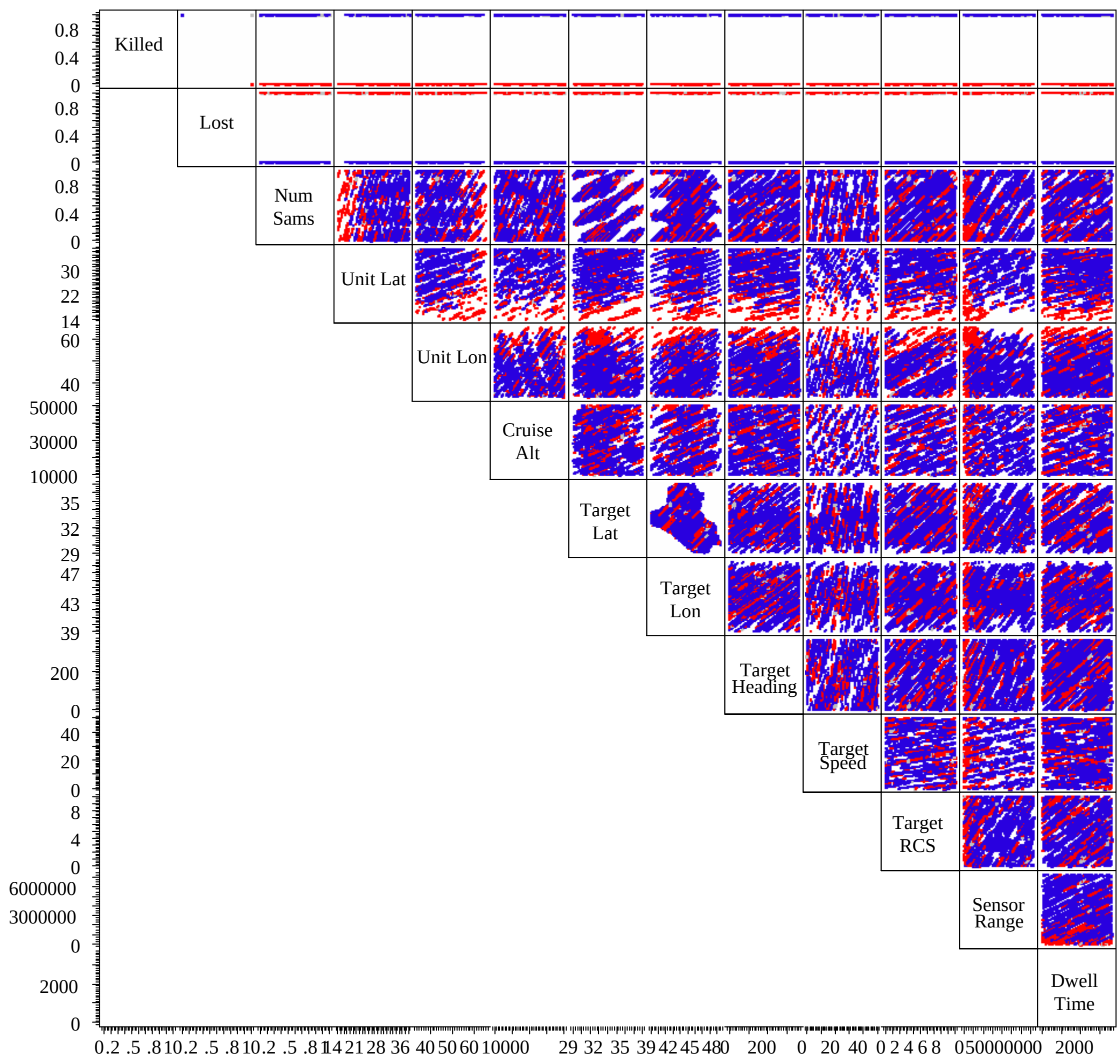


Figure 70: Multivariate Profiler for the Moving Target Training Experiment (2 of 2).

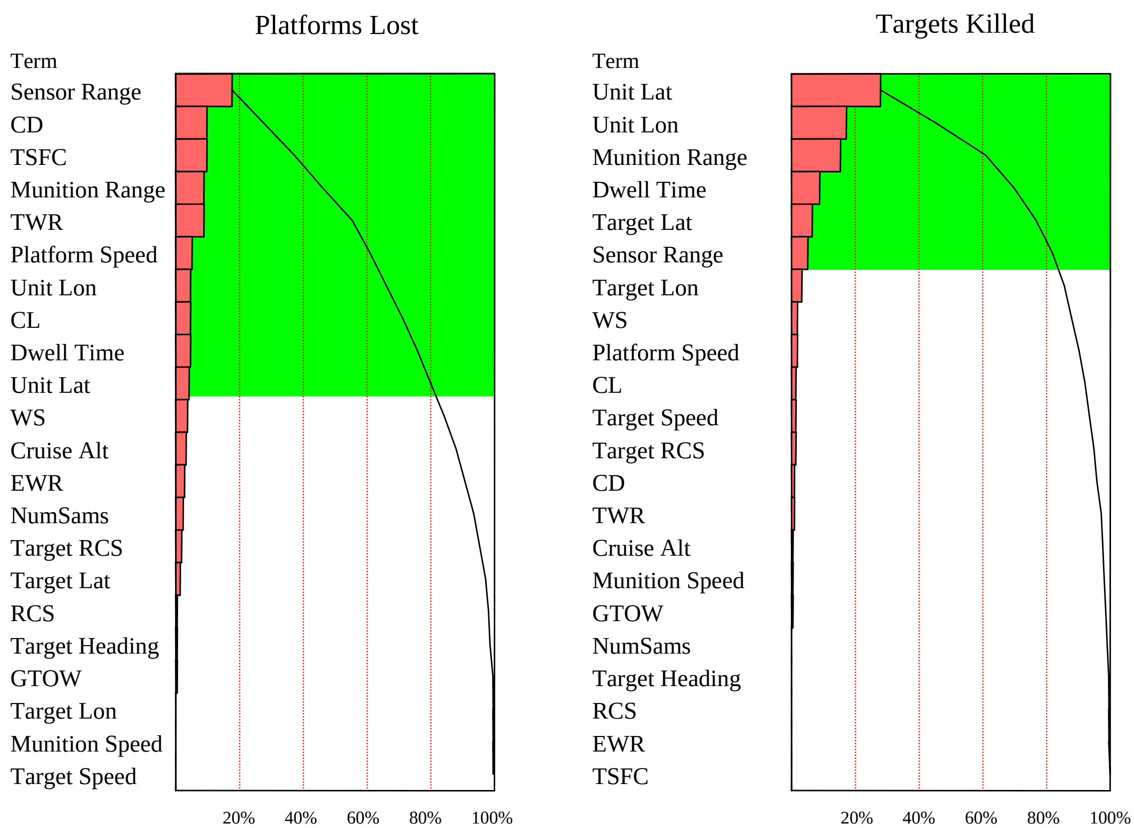


Figure 71: Pareto Plot Showing the Relative Influence of Design and Tactical Parameters on the Platforms Lost and Targets Killed Response.

The prediction profiler tool is useful in comparing partial derivatives as design variables are locked at different settings, but the trendlines are difficult to interpret for discrete responses. Instead, cases are partitioned into successful engagements (Targets Killed = 1, Platforms Lost = 0) and unsuccessful engagements (Targets Killed = 0, Platforms Lost = 1) and distributions of each subset are examined in turn. An example of one such distribution for sensor range is shown in Figure 72. Successful engagements tend to be distributed fairly uniformly throughout the span of sensor range while unsuccessful engagements are prevalent at short range. While such distributions are useful in identifying general trends, system-of-systems problems are dominated by the interactions between a variety of parameters. While the amount of data is overwhelming, only the multivariate profiler allows complete analysis of the total derivative, that is, an analysis of the change of responses while all input variables are simultaneously changing.

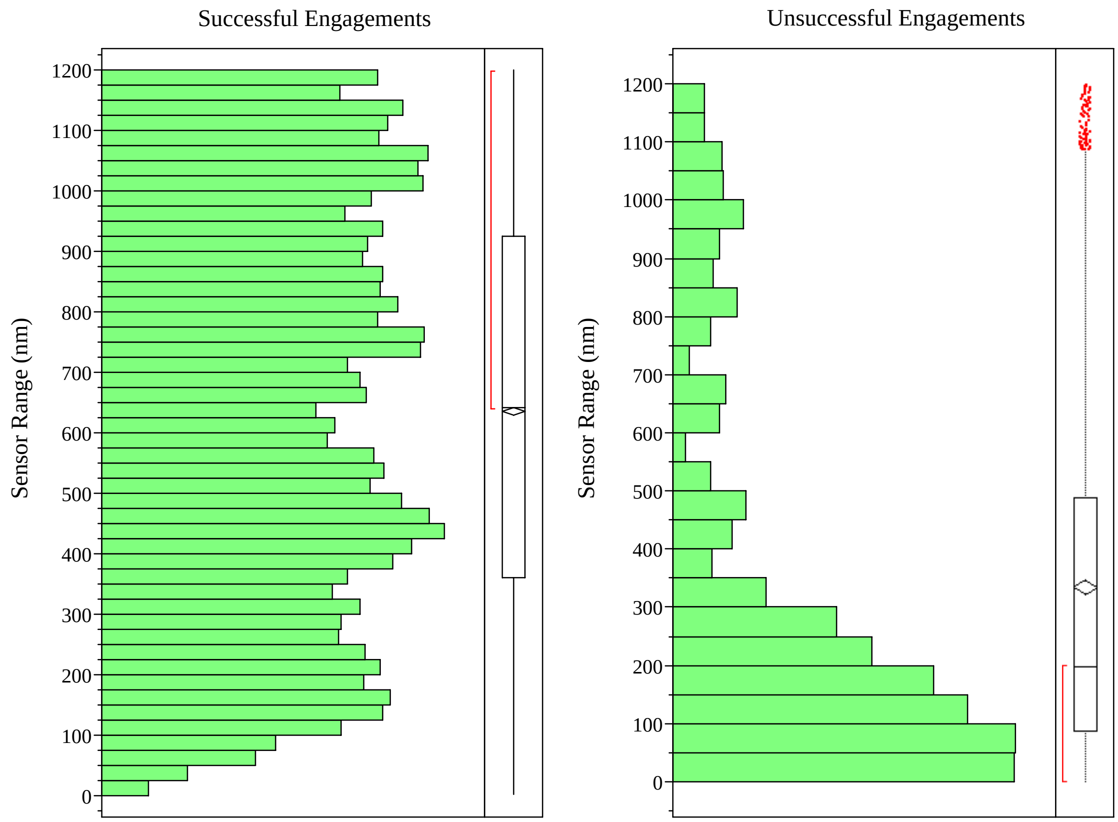


Figure 72: Analysis of the Sensor Range Distribution Reveals that Unsuccessful Engagements are Dominated by Low Sensor Range.

From analysis of the multivariate profiler, Pareto charts, and distribution plots, several trends can be discerned. Successful engagements were dominated by solutions where the platform was already located inside Iraq, while unsuccessful engagements were predominant along the southeastern edge of the Arabian Peninsula where the distance to theater was in excess of 1,852 km (1,000 nm). Long munition range and sensor range were key contributing factors to successful engagements. Holding these dominant factors constant would reveal the effects of other design parameters.

The data from the moving target experiment were analyzed using the BRAINN algorithm. Using the process outlined in Section 5.5.7, an initial exploratory pass was conducted using a lower training time and a range of topologies. The results of this phase are shown in Table 15. Using the optimum topology identified in this phase, a second set of regression was performed. The results for this phase are shown in Table 16. All measures of error for the two responses were below the 5% threshold identified as appropriate in Section 5.5.6. Based on the successful training of the battle manager for moving targets, the final form of the equation determined from the optimum pass was implemented inside the Time Critical Target Controller within the FLAMES simulation.

Table 15: Results from Battle Manager Neural Network Training for Moving Targets, Exploratory Phase.

	Response	
Parameter	**Platforms Lost**	**Targets Killed**
Validation Data (%)	25	25
Test Data (%)	3	3
Training Time (s)	300	300
Hidden Nodes (low)	6	6
Hidden Nodes (high)	15	15
Iterations at Each	20	20
Training % Correct	97.0587	98.5465
Validation % Correct	96.1437	97.6599
Test % Correct	95.7418	96.978
Optimal Nodes	8	11
Number of Cases	24,271	24,271

Table 16: Results from Battle Manager Neural Network Training for Moving Targets, Optimization Phase.

	Response	
Parameter	**Platforms Lost**	**Targets Killed**
Validation Data (%)	25	25
Test Data (%)	3	3
Training Time (s)	3600	3600
Hidden Nodes	8	11
Iterations at Each	12	12
Training % Correct	96.6638	98.7525
Validation % Correct	96.4239	97.4127
Test % Correct	96.2912	97.3901
Total % Correct	96.593	98.4385
Number of Cases	24,271	24,271

5.5.10 Evaluating the Effectiveness of Situations

In contrast to the scenario-level MoEs defined in Section 5.2.2, the MoEs for the one-on-one engagement are extremely simplified. Of primary interest is (1) whether the friendly bomber was killed and (2) whether the enemy target was killed for each experimental run. An additional metric of interest, the response time, can be calculated as:

$$ResponseTime = \frac{MunitionRange}{MunitionSpeed} + \left[\frac{(DistancetoTarget - MunitionRange)}{PlatformSpeed}\right] \quad (12)$$

Because the cognition model for the GITFCBomber is programmed to release the munition at its maximum range, the distance the platform flies is the difference between the starting range between the target and bomber (defined by geographic coordinates) and the distance flown by the munition.

To calculate the effectiveness of a given situation, a notional engagement cost is defined by Equation 13.

$$Cost = (T_K)TargetBonus - (B_K)BomberCost - (ResponseTime)TimeCost - MunitionCost \quad (13)$$

Where B_k and T_k are boolean parameters indicating whether the bomber and target were killed respectively. The BomberCost is a negative cost which penalizes the overall engagement cost for each bomber lost. Using cost estimates for the JSF, F-22A, and B-2A as a function of weight, a simple linear regression yields with high confidence (R^2 of 0.99975) a relationship of:

$$BomberCost_{(\$M)} = \left(5.8442843572e^{-3}\right)\left(Weight_{(lbs)}\right) - 199.58_{(\$M)} \quad (14)$$

Which provides a *rough* dollar value to penalize the overall cost function[20]. The FLAMES platform model requires empty weight as an input and calculates gross weight as the sum of fuel weight and empty weight. Since the fuel weight changes throughout the mission, this is not a reliable way to calculate platform cost. Equation 14 can be written in terms of empty weight (R^2 of 0.9989) as:

$$BomberCost_{(\$M)} = \left(1.41571e^{-3}\right)\left(EmptyWeight_{(lbs)}\right) - 234.362_{(\$M)} \quad (15)$$

[20]Stealth aircraft assumed.

To calculate the cost penalty for a munition fired, a neural network surrogate model was created for eight munitions for which weight, range, and cost information could be obtained from references [106] and [218]. These munitions provide a rough estimate of the cost of munitions across the spectrum from guided bombs to long range cruise missiles. The values used are shown in Table 17.

Table 17: Parameters Used to Calculate the Munition Penalty Cost [106, 218].

Munition	Penetrating?	Weight (kg)	Range (km)	Cost
MK-82	No	227	5	$9,000
MK-84	No	907	5	$22,000
AGM-65B	No	462	8	$64,100
AGM-65D	No	484	20	$111,000
AGM-65G	No	675	25	$269,000
AGM-84B	No	1386	95	$346,000
CALCM	No	1474	1200	$1,500,000
UGM-109C	No	1315	1250	$1,100,000

A cost of $100,000 per hour was applied to the *TimeCost* to illustrate the benefit of speed for the GSTF mission. Finally, the "cost" of the target was defined at $20,000,000 to provide a bonus for attacking the target. In this case, the value of the target exceeds the cost of all munitions while it is much lower than the cost of the platform. This means that no target is worth losing a platform over and all targets are worth a munition. While this situation may not always be true, alternative settings may teach the intelligent battle manager when a better option is not to engage the target at all or when a suicide mission is justified. The above doctrine is guided by observations from past conflict that place an extreme value on platforms and pilot lives.

According to Garner's review of Operation *Desert Storm*, the estimated cost of the conflict was approximately $56B (1991 USD). During the first 24 hours, the U.S. launched approximately 100 cruise missiles ($100M), 500 HARM/Shrike anti-radiation missiles ($50M), 20,000 tons of bombs ($200M), expended $50M in fuel for the 1,000 participating aircraft and incurred approximately $100M in cost due to damaged and destroyed aircraft [162]. The primary objective of assigning a cost to each engagement is to place emphasis on using highly capable and expensive weapons when necessary, but to switch to equally capable and less expensive munitions as conditions warrant. Future work will examine economic trades in more detail.

5.5.11 When is it Appropriate to Use a Meta-General?

The cognitive approach highlighted above is only one of the model paradigms for modeling human or organizational behaviors. In 1998, the National Research Council performed an overarching study of existing and proposed paradigms as well as their applicable domains of application [307]. Oeltjen notes that a hybrid approach that synthesizes elements of these paradigms is often needed to create computational cognitive models for military simulations [324].

While the Meta-General is useful for identifying platform/weapon combinations based on simple knowledge of strategic decisions, the technique is resource intensive for relatively simple decisions when the same answers are always true regardless of the state of the scenario. The best example of such a decision is the allocation of weapons against hardened targets: striking a hardened target with a weapon that has a zero probability of kill for that type of target wastes a strike that could otherwise have been allocated to a worthwhile mission. Adding the property of weapon hardness and target hardness to the battle manager training algorithm would have resulted in the addition of two variables to the DOE when the underlying logic in the strike function dictates that weapons whose capabilities do not exceed a target's hardness requirements are non-functional. To avoid unnecessary code execution, a simple check statement was added to the battle manager's target allocation routine.

Another example is the allocation of platforms to attack time critical targets. If a detected target has a maximum speed greater than zero, it is capable of movement and it is assigned using a Time Critical Target Controller. If, on the other hand, the maximum speed is zero, the target is assigned using a Ground Controller. In these types of situations, a knowledge-based rule set is the most appropriate cognition method.

5.5.12 Combining the Meta-General with Knowledge-Based Rule Sets

To reinforce this point, the trained Meta-General was combined with a knowledge-based rule set for target hardness properties for the reasons described above. While this choice has an operational benefit in terms of reduced computational time, it also demonstrates that the Meta-General approach can be combined with existing hard-coded rule sets where appropriate: implementing the SOCRATES methodology on a legacy code does not necessarily require rewriting the entire simulation. The following experiment demonstrates that the techniques in this work can be applied in a useful manner to existing simulation efforts.

Just as the properties of each target set were assigned to individual targets (see Section 5.4.1), the pseudo-signature technique can be used to identify weapon and target hardness. A property called "target hardness" can be defined by using a pseudo-signature at 8,000 MHz with RCS values of 1.0, 3.0, and 10.0 to represent low, medium, and high hardening respectively. Using the signature selector (shown in Figure 52), the hardness property can be assigned to both the platform to be attacked and the munition to be used in the attack. A routine was added to the GITFCGroundController's assign function that queries the hardness pseudo-signature of the platform to be attacked. This routine compares the hardness value of the target to the RCS value of the munitions on the blue platform preferred by the battle manager for this engagement. If the hardness value of the blue munition is greater than or equal to the hardness value of the target, the platform is assigned. Otherwise, the platform is skipped and the next most desirable platform identified by the Meta-General is checked for this compatibility.

If no subordinates assigned to a ground controller can accept the engagement the battle manager is notified and tries to reassign the target to a different ground controller. Future modifications will allow the battle manager can add the desired munitions to an idle platform and generate the sortie if both weapons and aircraft are available. When such a sortie is generated, a time delay is added to the platform to represent the time it would take to reconfigure the aircraft on the ground[21].

[21] Future work may examine the sensitivity of the results to the variability of ground events and logistical concerns.

5.5.13 Implementing the Meta-General in FLAMES

The surrogate model developed in the subsequent sections are mathematical equations that require as inputs (1) the location of an enemy target (2) the location of a friendly asset and (3) the physical parameters of the friendly asset in question. The output of these equations is the expected cost of the engagement as given in Equation 13. The neural network equations were coded within the GITFCGroundController and the GITTCTController cognition models for stationary and time-critical targets respectively. For each detected target, the latitude and longitude can be queried from the FLAMES kernel. This specifies the fixed input values for the neural network. Next, the battle manager iterates through all available platforms under its control. For each platform, the latitude and longitude of its current position are known. By querying the platform and its weapon system, it is possible to obtain the current weight, maximum speed, munition speed, munition range, and other parameters required to evaluate the neural network equations. With this information, the inputs to the equations are completely defined and the expected cost of the engagement can be calculated. After iterating through the list of available platforms, provided that the engagement can be completed successfully by one or more platforms, the platform with the lowest expected cost is tasked to take the engagement. If the battle manager determines that the engagement cannot be successfully completed by any platforms under its control, it ignores this target and proceeds to the next target on the prioritized list. As the simulation continues, the original target remains on the top of the priority list. When a suitable platform becomes available or when the density of SAM defenses drops below a dangerous level, a platform is automatically assigned to destroy this target since it remains the highest priority target in the ranked list.

After being tasked, the individual agent performs additional checks to see if it can complete the engagement with a high probability of success. The development of the on-board intelligence to enable realistic decentralized execution and exploitation of technologies is described in the subsequent section.

5.6 *Step 6: Create Intelligent Agents Using Response Surface Equations*

> *"So a military force has no constant formation, water has no constant shape: the ability to gain victory by changing and adapting according to the opponent is called genius."*
>
> -Sun Tzu

While the Meta-General supports the Air Force's weaponeering function, the doctrine of centralized control and decentralized execution avoids micromanagement of flight plans and tactical employment of air power. According to Air Force doctrine, "the tactical level of aerospace warfare deals with how ... packaged forces are employed and the specifics of how engagements are conducted and targets attacked" [430]. While the targeting function (Section 5.4.1) primarily addresses "why" we fight, and the weaponeering function (Section 5.5) addresses "what" we fight with, the tactical employment of airpower deals with "how" we fight [430]. The role of tactical employment and mission planning in the Air Force target cycle is shown in Figure 73.

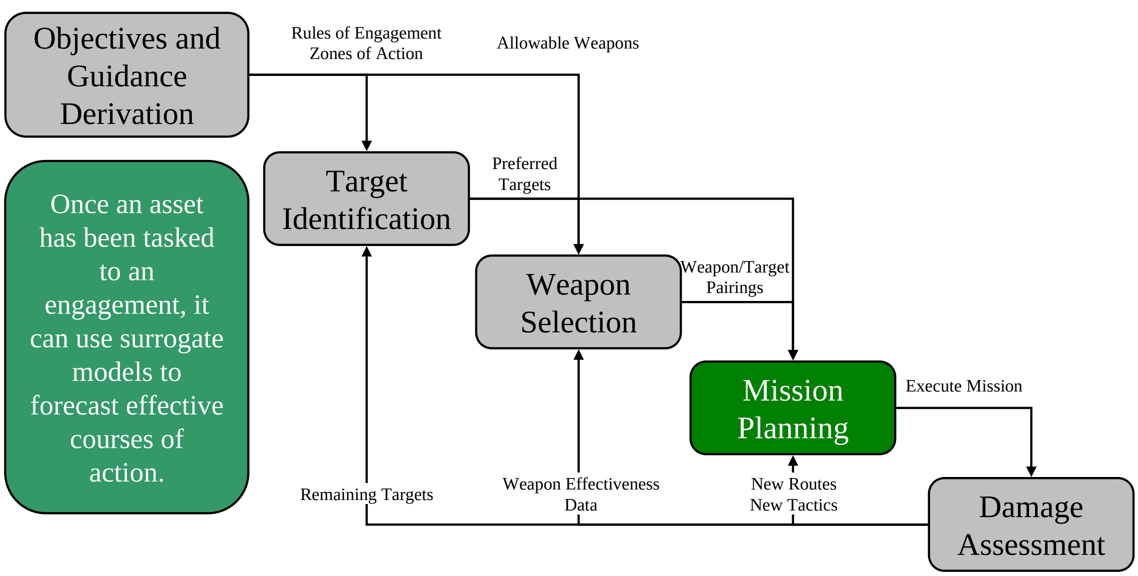


Figure 73: The Role of Mission Planning in the Target Cycle (Adapted from Reference [429]).

The mission planning phase requires as inputs the list of targets to attack and the preferred pairings from the battle manager. The product is a mission plan, which is executed until one or more completion criteria are reached.

The purpose of the battle manager is to formulate the Air Tasking Order (ATO), which identifies "projected sorties, capabilities and/or forces to targets and specific missions," the "supporting commanders execute the ATO as tasked and recommend changes as appropriate" [468, 430]. Under this directive, the actual employment of aerospace power and determination of effective courses of action falls to the individual air wings and eventually to the pilots themselves. Therefore, within the simulation, the weapon selection phase is performed by a single battle manager and the mission planning phase is conducted by each individual agent that is tasked for a mission.

5.6.1 Using a Playbook of Tactical Options

While some techniques in artificial intelligence and machine learning focus on the development of new maneuvers or tactics based on real-time learning and adaptation, such behaviors are difficult to quantify for multi-agent systems where the evolving behaviors of individual agents confound the actions of each other. A workshop held in 2002 by the Neural Information Processing Systems Foundation noted that "multi-agent learning poses significant theoretical challenges, particularly in understanding how agents can learn and adapt in the presence of other agents that are simultaneously learning and adapting" [25]. Paradigms under development to address this issue include Q-learning, game theoretical approaches, network optimization, and reinforcement learning. While Section A.2.7 outlines some of these emerging techniques, most of the research in this field has been confined to simple games and AI-based learning which has not been synthesized into a practical application to this problem domain as of yet. Furthermore, in the short time frame of hours to days that the GSTF operates in, pilots are unlikely to adapt new tactics for which they have not been trained. For these reasons, the proof-of-concept exercise primarily focuses on the ability of intelligent agents to synthesize tactics from existing basic maneuvers provided in a "playbook" of options that are defined *a priori*.

Examples of potential tactical options include the ingress altitude, flight speed, route to target, refueling options, and distance to target at the point of weapon release. The space of potential tactical option defines the allowable "plays" from which the CONOPS of a particular agent are derived. The choice of different tactical options depends greatly on the properties of the intelligent agent, which often rely on technologies. As Lambeth notes:

> *"low observability further ensures an increase in a stealthy aircraft's effective operating radius, owing to its capacity for enabling the aircraft to operate at fuel-efficient speeds and altitudes and for reducing the need for weight-adding and space-occupying electronic countermeasures that otherwise would be required for self-protection against enemy defenses"* [253].

While decision trees and logic gates provide a means to establish the heuristics of the playbook, a method is needed to rapidly evaluate which "plays" are preferred under different operational conditions and in the presence of new technologies.

5.6.2 Response Surface Equations Can Be Used to Calculate Performance Metrics

As noted above, the selection of a particular tactical maneuver depends on the properties of the agent for which the maneuver is employed. For example, height, weight, and build are three properties that can be attributed to a person. The person's ability to swim 100 meters in a specific time may be a function of these three parameters. Similarly, the ability of an aircraft to perform specific tactical maneuvers is constrained by one or more design/operational parameters. A method is desired to related the design/operational parameters to quantitative measures of performance for the given aircraft. By establishing thresholds based on these measures of performance for each "play" in the playbook, the allowable tactical maneuvers based on the state of the agent may be defined. One way to realize this objective is by providing surrogate models to each intelligent agent.

Section B.3 demonstrates a proof-of-concept of this technique using a Performance Vector of Attributes (PVA) for an air-to-air combat scenario. In a LRS scenario, however, this objective function is not as important because a long-range bomber does not typically engage in air-to-air combat nor is it designed to do so. Two primary measures of performance, survivability and range, can be used to define the playbook constraints for an LRS aircraft.

Once a target has been identified, the straight-line distance between this target and any blue aircraft can be calculated using the FLAMES kernel; however, this distance is of little use. Unless an aircraft possesses unusually high survivability characteristics, a friendly aircraft would choose to avoid unnecessarily high concentrations of SAM sites and plot a minimum threat "blue line" around hostile air defenses[22]. To simulate such paths, a series of airspace corridors are defined in the FLAMES framework as shown in Figure 74.

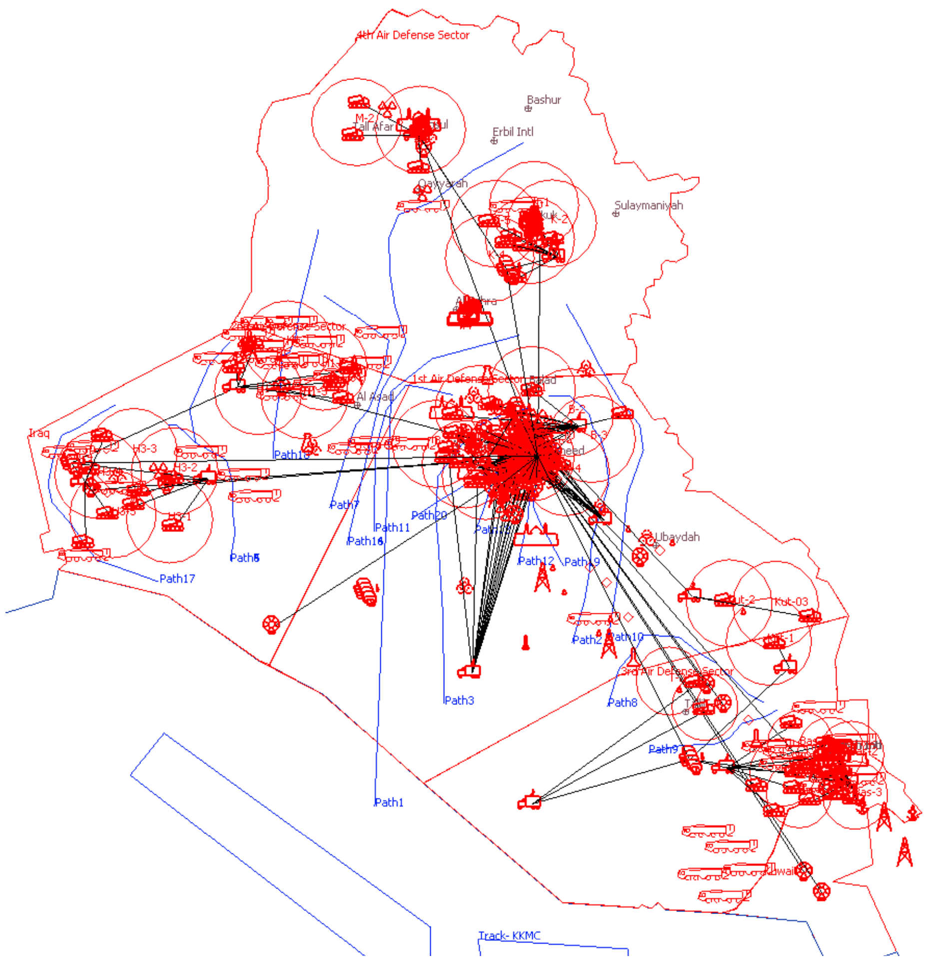


Figure 74: Airspace Corridors in the FLAMES Simulation.

[22]The creation of this path assumes that the locations of SAM sites are known. For stealth aircraft, "mission planners [construct] a rigid flight plan known as the 'blue line,' designed to optimize both the [aircraft's] low-observable characteristics and the support available from jamming, defense suppression, and air superiority aircraft" [186].

Twenty different pathways are illustrated in Figure 74. These notional routes are partially based on operational data from the Gulf War Air Power Survey [105]. The length of each route can be calculated in FLAMES by calculating the vector distance between each point in the corridor. Available airspaces are partially constrained by relating this length to the range of a platform, taking into account locations of nearby bases and refueling aircraft. Additionally, conditions for entry into each corridor may be constrained by the platform RCS and the speed of the platform. As mentioned in Section 5.2.1.5, the air defense environment of the Iraqi theater makes it impossible to operate below 10,000 feet. Several "high-risk" airspaces that shortcut through densely defended regions allow flight below this level if the speed of the platform is sufficiently high to avoid dense anti-aircraft artillery (AAA) and man-portable air defenses (MANPADS).

An algorithm developed by Tangen iterates through available flight path options and identifies intersections with active SAM sites based on missile range [394]. This calculation is used to determine the threat level of each potential option. Then, the agent uses a random search to query a surrogate model at multiple altitudes and speeds for each airspace to determine the optimum flight path that maximizes the probability of successful mission completion. A demonstration of this technique using the Operation *Desert Storm* scenario is described in the subsequent sections.

When contrasted with the straight-line vector calculations, significant increases in survivability are observed when the route planning algorithm is used. In an example ten-hour engagement using notional LRS platforms, all platforms were lost when straight-line vectoring was used. When this scenario was replayed using the twenty airspace corridors shown in Figure 74 and intelligent agents, no platforms were lost in the ten hour time span.

5.6.3 Demonstration of Intelligent Agents Using Surrogate Models

Since the choice of allowable airspace regions is partially dictated by range and speed, a method is needed to rapidly calculate both parameters at runtime to evaluate which airspace corridors may be used for an attack. The process diagram shown in Figure 75.

The first objective is to calculate the platform speed. First, several platform design

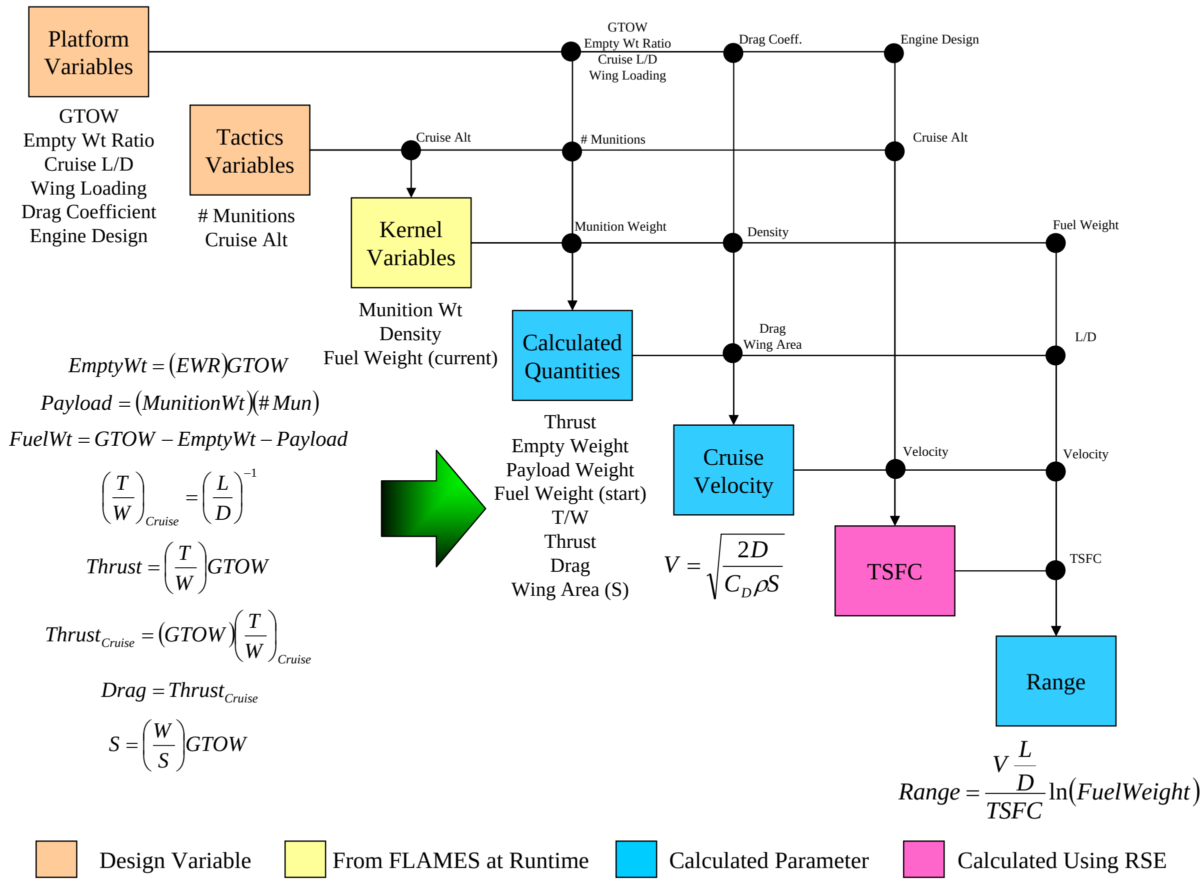


Figure 75: A Process for Calculating Measures of Performance Using Surrogate Models at the Agent Level.

variables including the gross takeoff weight (GTOW), the empty weight ratio, and the number of munitions can be used in conjunction with runtime variables such as the current fuel and munition weights to calculate the weight of a platform at run time. Also, for straight and level flight, the thrust-to-weight ratio is the inverse of the lift-to-drag ratio which is also a defined input parameter. Given these inputs, the cruise thrust-to-weight (T/W) ratio of the platform can be used to calculate thrust using the equation:

$$T = \left(\frac{T}{W}\right) W \tag{16}$$

Also, noting that thrust and drag are equal for straight and level flight, the equation for drag (D) can be solved in terms of velocity:

$$V = \sqrt{\frac{2D}{\rho C_D S}} \tag{17}$$

Where the density, ρ, is a function of altitude and can be found by querying the atmosphere record in the FLAMES kernel and the drag coefficient, C_D, and wing area, S, are design parameters of the aircraft that can be found by querying the platform. Cruise velocity is then used as an input to the propulsion system surrogate model and the range estimate.

5.6.3.1 Development of a Surrogate Model for Propulsion Systems

To calculate platform range, it is necessary to provide an accurate estimate of the fuel burn over the remaining distance to and from the target area. To address this issue, a surrogate model that calculates TSFC as a function of altitude and velocity is needed. Altitude can be determined by querying the atmosphere record in the FLAMES kernel, and a process for defining cruise velocity was defined in the previous section. These values define the "state" of the agent at a position in space. When combined with the design variables for a propulsion system, a thermodynamic model that calculates TSFC can be created.

While a number of thermodynamic cycle codes are available to calculate engine performance including the Numerical Propulsion System Simulation (NPSS), the NASA Engine Performance Program (NEPP), and commercial codes such as GasTurb, a simple and robust model for calculating basic properties is the Thermodynamic Properties (THPROP) analysis tool [314, 242, 252, 469]. THPROP is a series of Microsoft® Visual Basic routines that calculate the properties of basic thermodynamic functions such as compression, expansion, and combustion. When the routines of THPROP are appropriately configured, a designer can model propulsion system architecture such as turbofans, turbojets, and ramjets. A Microsoft® Excel interface to the THPROP routines is shown in Figure 76.

The original THPROP routine was not well-suited for a design of experiments: the worksheet is extremely unstable and crashes when unrealistic combinations of efficiencies, temperatures, and pressure ratios are applied. Error trapping routines were added to the THPROP routines to support the execution of a DOE.

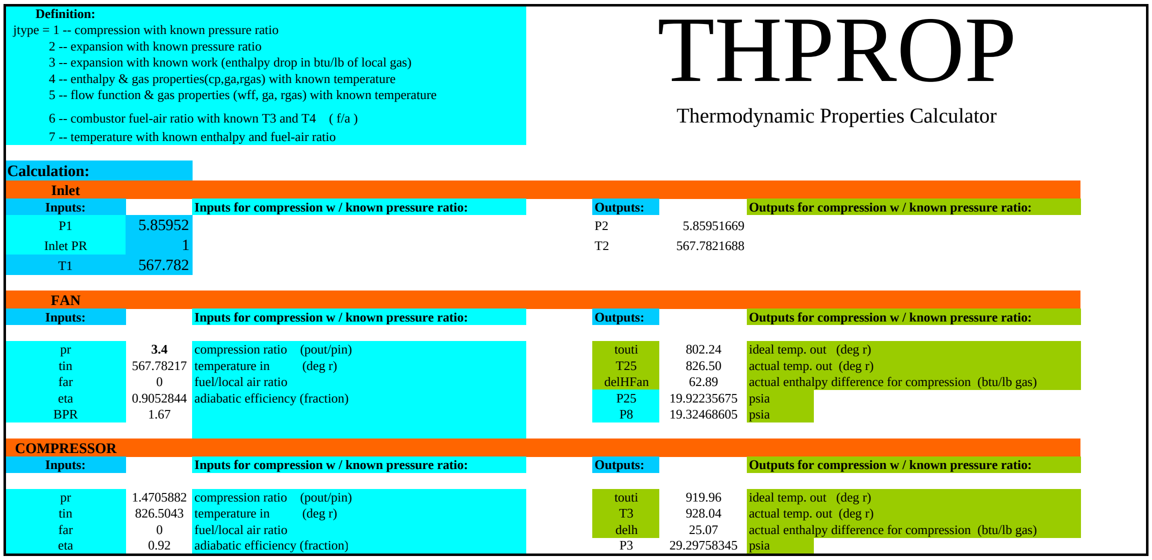


Figure 76: THPROP Thermodynamic Properties Calculator for Propulsion System Architectures.

In 2005, Engler demonstrated the viability of using neural network surrogate models to approximate the performance space of a turbofan, turbojet, and ramjet engine using GasTurb [143]. Based on the success of this approach and the limited applicability of polynomial surrogates to the engine problem *when altitude and speed are input parameters*, a neural network was selected as an appropriate type of surrogate for this demonstration. This defines the need for a large space-filling design to provide the necessary data needed for regression. The rapid run time of the spreadsheet model does not preclude this approach. Using the MATLAB® Model Based Callibration toolbox, a 35,000 case space-filling design was created. The ranges of the DOE are shown in Table 18.

Table 18: Ranges for the Design of Experiments for Propulsion System Surrogate Models.

Parameter	**Low**	**High**
Mach Number	0.75	5
Altitude, m (ft)	0	16,000 (52,493)
Fan Pressure Ratio	1	4
Fan Efficiency	88%	94%
HPC Efficiency	88%	94%
Bypass Ratio	0	4
Overall Pressure Ratio	5	25
Turbine Inlet Temperature, °R (°K)	2000 (1111)	4000 (2222)
HPT Efficiency	88%	92%
LPT Efficiency	88%	92%

The 35,000 case DOE was executed on a single Pentium IV 3.0 GHz computer in approximately three hours. Error trapping features added to the THPROP routines prevented program crashes during execution, and resulted in identification of 25,719 viable data points for regression. Of these, 4,078 resulted in negative TSFC and 18,650 resulted in negative specific net thrust. This behavior was primarily driven by combinations when the exit temperature of the compressor was greater than the design turbine inlet temperature, resulting in a negative fuel/air ratio or when the efficiency values for the turbine components were too low to provide enough work to drive the fan and compressor. While these cases result in infeasible engine solutions, the data is still valuable in the creation of a neural net: the network models these infeasible values which can be used to understand both viable and non-viable design spaces.

Fitting the TSFC response proved extremely difficult with the full data set: failed cases could have TSFC values as low as -6390 and as high as 143.6 lb_m/lb_f-hr. While it was possible to fit a neural network to all the points in the data set and capture the behavior of failed cases, the error in the region of interest between 0 and 2 lb_m/lb_f-hr was as high as 50%. To address this issue, a two-phased approach was used to regress TSFC. First, the response for the specific gross thrust of the core (SFG Core) was converted to a boolean where zero indicated an engine configuration that was infeasible and unity corresponded to a valid engine configuration. A nine hidden node neural net with a training percent correct of 99.973%, a validation percent correct of 99.8755% and a test set percent correct of 100% was created from the SFG Core response using the BRAINN tool. Then, cases with a negative SFG Core were removed from the TSFC data set.

The BRAINN tool was used to regress the 25,719 data points for the responses of specific net thrust (SFN) and the 11,770 valid data points for specific fuel consumption (TSFC). An exploratory pass identified 15 hidden nodes and 16 hidden nodes as the optimum topology, respectively. A second training pass was conducted using five iterations with a 3600 second training time at the optimum node setting. The resulting goodness of fit metrics of the two responses are shown in Table 19.

While high-thrust engines like the Pratt and Whitney F-119 that powers that F-22A have demonstrated the ability to power a supersonic cruise up to approximately Mach 1.7 without using afterburners, it is anticipated that an afterburner would be required for long duration flight above Mach 2.0. While the thermodynamic model converges to high speed solutions without an afterburner, these architectures are not realistic. Therefore, the above process was also performed for a turbofan engine architecture that featured an afterburner, with an allowable temperature from 2000°R to 4000°R (1111 °K to 2222 °K). A fifteen node neural network of the SFG boolean response was created with 100% training percent correct, 99.8613% validation percent correct, and 99.6146% test set correct. The goodness of fit metrics for the SFN and TSFC responses for both engine architectures are shown in Table 19.

Table 19: Goodness of Fit Metrics for the Neural Network Equations for the Turbofan Propulsion System.

	No Afterburner		**Afterburner**	
Parameter	**SFN**	**TSFC**	**SFN**	**TSFC**
Validation %	25	25	25	25
Test %	3	3	3	3
Training Time (s)	3600	3600	1800	1800
Iterations	5	5	3	3
R^2 Training	99.8243	99.4956	99.8140	99.9399
R^2 Validation	99.8079	99.2020	99.7683	99.1800
R^2 Test	99.8371	99.0535	98.7440	99.8747
Optimum Number of Nodes	15	16	14	15
Number of Points	25,719	11,770	7,836	7,836

In FLAMES, Neural networks from both the SFG Core and TSFC are required to evaluate TSFC: first, the input data is used to evaluate whether SFG Core is unity, and if so, then the second neural network is used to evaluate the specific value of TSFC. In this manner, an intelligent agent can identify flight regimes where its propulsion system does not work using the first neural net, and if the propulsion system is functional, it then uses the second neural net to evaluate how good the fuel consumption is based on the technologies applied. If the propulsion system is not efficient in the airspace corridor that the agent is trying to fly through, the agent needs to change altitudes or select a different route. Logic to enable these decisions was coded in FLAMES within the platform “fly” routine and engine subsystem definition routine.

The neural network equation for TSFC was used to formulate a range estimate for each platform, as defined in the subsequent section.

5.6.3.2 Development of a Surrogate Model for Platform Range

When a platform is tasked to engage a target, it selects the "best" route to ingress to the target area and egresses along the same route. The distance from the current position to the nearest point of the desired route, the length of the route, and the distance from the exit of the route to the weapon release point is used to calculate the necessary range of the platform. Surrogate models that calculate speed are used to constrain entry into "high-risk" airspace corridors while surrogate models for range are used to assess the ability of a platform to complete its mission with available fuel reserves.

While a number of vehicle synthesis and sizing routines such as NASA's FLight Optimization System (FLOPS) and AirCraft SYNThesis (ACSYNT)[23] or DARcorporation's Advanced Aircraft Analysis (AAA), a simple surrogate model for aircraft sizing is the Breguet range equation for jet aircraft[24]:

$$Range = \frac{V\frac{L}{D}}{TSFC}\ln\left(\frac{W_{initial}}{W_{final}}\right) \tag{18}$$

As shown in Figure 75, this equation requires as inputs the velocity, which is calculated from the assumption of straight and level flight, the lift-to-drag ratio (which is the inverse of the thrust-to-weight ratio for straight and level flight), the empty weight (a design parameter), the weight, which can be determined from the FLAMES kernel, and the TSFC. The development of a surrogate model for the calculation of TSFC is summarized in Section 5.6.3.1.

A multivariate plot showing a 10,000 case exploration of the sizing surrogate described in Figure 75 is shown in Figure 77. To obtain a first-order impression of the design and requirements sensitivity, a filtered Monte Carlo technique can be applied to examine regions

[23] Now marketed as a commercial product, ACS, by Avid, LLC.

[24] The first known publication of a range equation for piston aircraft was published by Coffin in 1919 [103]. The same equation was later independently derived by Louis Charles Breguet in 1923 and came to be known as the "Breguet Range Equation" [73]. Reference [57] reviews the development of range equations for jet aircraft, a variant of which is used herein.

of interest for LRS design studies.

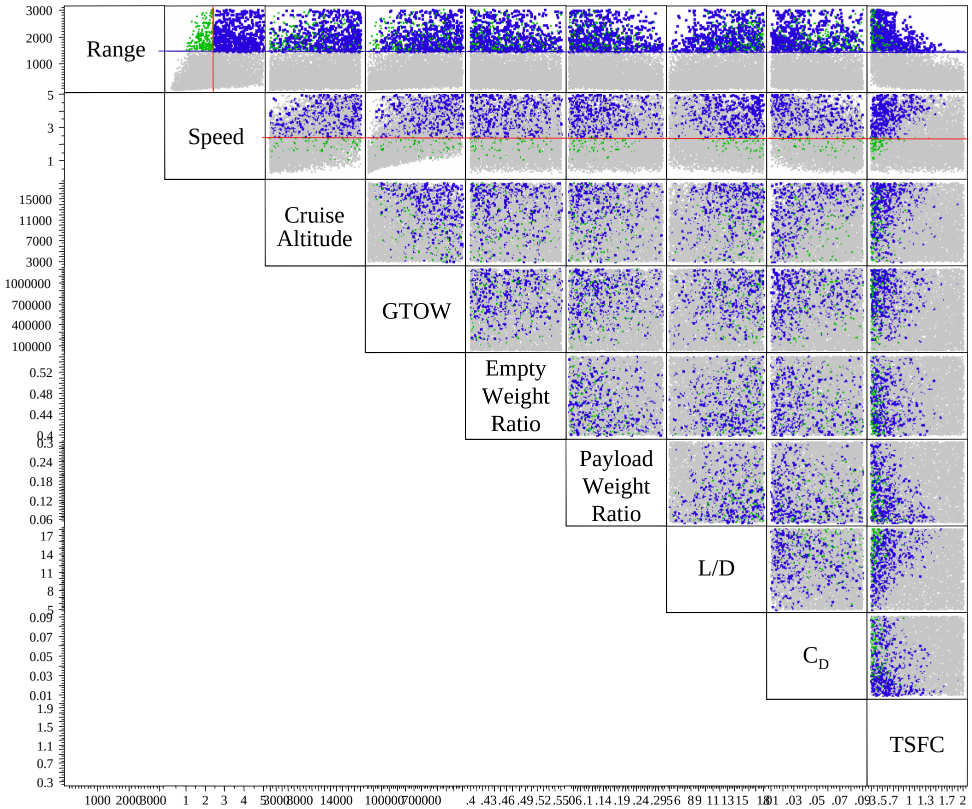


Figure 77: Multivariate Profiler Showing 10,000 Cases Analyzed Using the Breguet Range Equation Formulation.

Based on the requirements exploration for LRS concepts in Section 2.4, the upper bound for platform range is established at 5,556 km (3,000 nm). The blue line in the range box indicates a constraint of 2,778 km (1,500 nm) established by Shaplak as a reasonable range constraint to operate on the Arabian Peninsula [373]. Speeds in excess of Mach 5 were eliminated from consideration. The red line in the speed box indicates a constraint of Mach 2.4, one of the speed boundaries that change the definition of LRS concepts according to Watts [478]. The green points are indicative of solutions with a range greater than 2,778 km and a speed less than Mach 2.4, while the blue points identify solutions with a range greater than 2,778 km and a speed greater than Mach 2.4. The gray points identify other solutions across the spectrum of cases examined.

5.6.3.3 *Implementation: Using Surrogate Models to Provide Tunable Cognition*

Due to its simplicity, it is not appropriate to model the Breguet equation with an RSE or neural net since the first tends to under-fit and the second tends to over-fit the equation. Equation 18 was coded directly into the cognition model for route planning. The surrogate model for TSFC and the Breguet equation for range were added to the *vector to target* function of the bomber cognition model. For each of the twenty available routes, the current platform range is compared to the vector distance through each corridor to the assigned target. If no path can be successfully navigated based on the available fuel load, the platform attempts to rendezvous with a tanker closer to the corridor entry point and decline the engagement. This makes the target available for tasking to other platforms. Once the refueling operation is completed, the platform again becomes available and may be reassigned a new target. The process repeats and the platform continues to evaluate potential ingress routes against its fuel load until weapons are depleted or the scenario concludes.

In addition to calculating the distance along the corridor to the target, the platform also calculates the "threat density" along the airspace by adding up the number of SAM sites within firing range of the airspace corridor. This calculation is a simplification, because in reality the pilot would not have access to such information in real time. The best assessment of active SAM locations would be provided during a pre-flight briefing by intelligence officers [436].

Using thresholds on response time and threat density, the intelligent agent "decides" which path to choose. If a desired path is too dangerous, the agent may reevaluate this path later in the campaign after SAMs have been suppressed and the threat density in the area decreases. This behavior is illustrated in Figure 78. As illustrated in the figure, during the first two hours, only paths 2, 8 and 11 are chosen for ingress. Six hours into the conflict, path 14 becomes desirable due to the proximity to certain targets and the reduction in the overall threat level. As the simulation continues, the apportionment of engagements across different corridors changes as the agent decisions are influenced by a changing distribution of potential targets and threats. The distribution of corridor choices is also influenced by

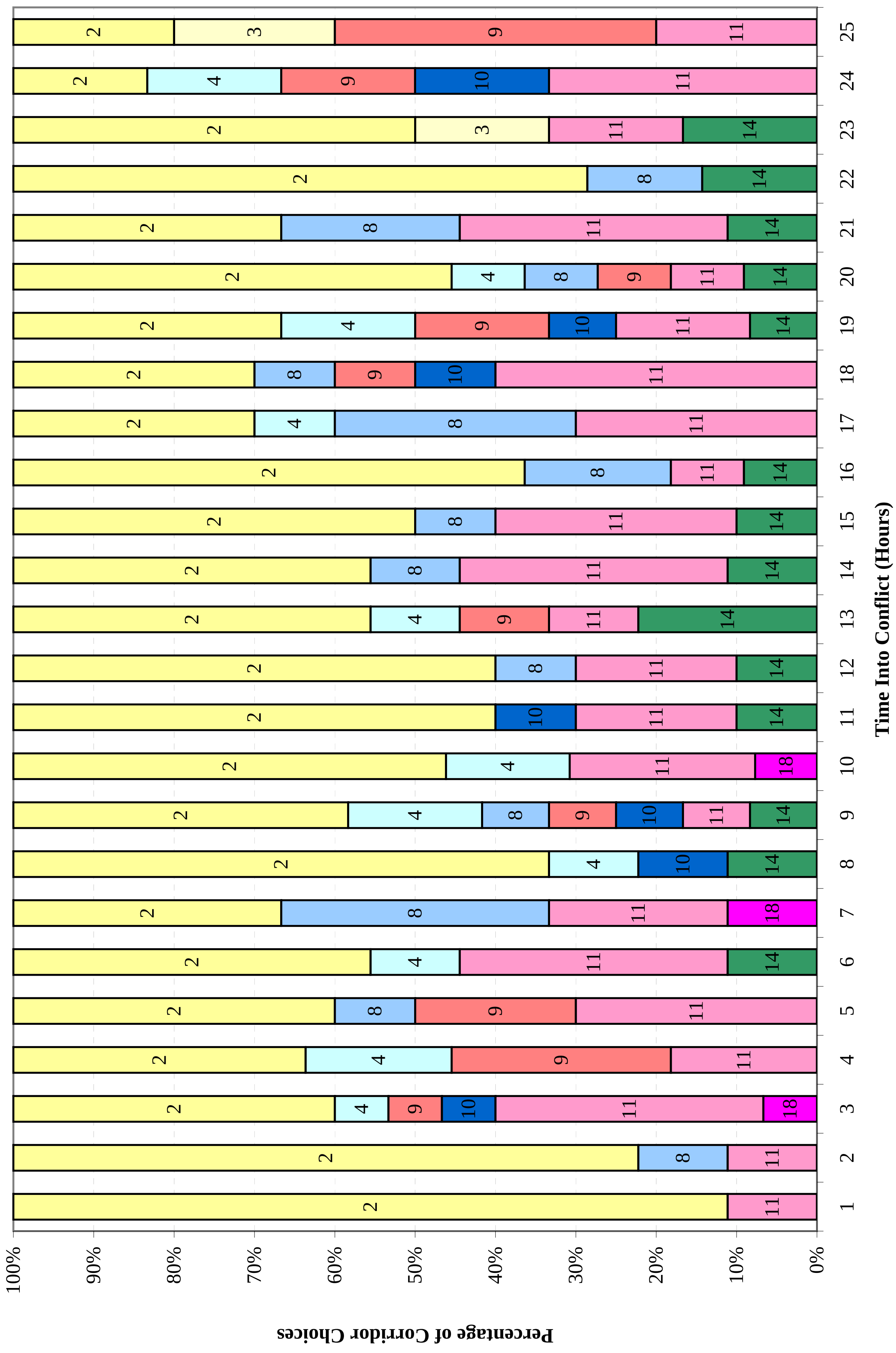
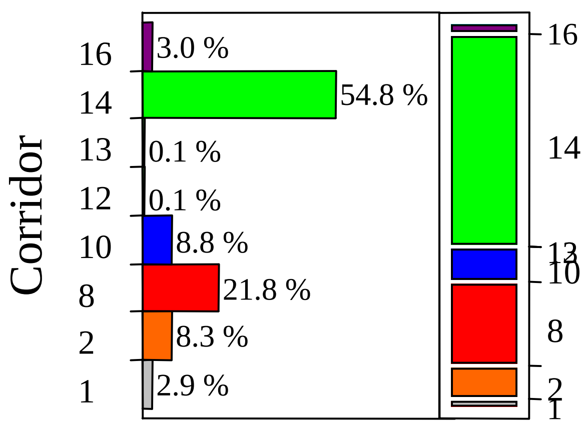


Figure 78: Analysis of Preferred Airspace Corridor Selections Over Time.

a change in the radar cross section of the platform and an alteration in the decision logic that triggers different cognitive behaviors.

The previous paragraph and supporting data in Figure 78 demonstrate how an intelligent agent changes behavior as its perception of the threat environment changes for a given aircraft configuration. To demonstrate how an agent can be "tuned" to make different decisions, another experiment was performed in which a single engagement was evaluated in the presence of different technologies at the aircraft and engine level. Using the neural network surrogate models for the afterburning and non-afterburning engine previously discussed and the Breguet range equation, a simulation was executed where the aircraft and engine were varied parametrically and the preferred airspace corridors for the engagement were tracked. A summary of the desired routes through the twenty pre-defined airspace corridors is shown in Figure 79.



Corridor #	Count	Probability
1	670	0.02930
2	1,909	0.08349
8	4,995	0.21847
10	2,014	0.08809
12	33	0.00144
13	29	0.00127
14	12,533	0.54815
16	681	0.02978

Figure 79: Analysis of Preferred Routes Through Airspace Corridors for a Platform with Variable Technologies and a Fixed Threat.

In general, the agent prefers to ingress through corridor 14; however, there are conditions where it chooses from a group of other corridors. The multivariate plot shown in Figure 80 shows the correlation between the corridor choices and aircraft/engine design variables by color-coding the different corridors selected. Discrete color bands are clearly visible in the RCS row. According to Figure 80, when the RCS is between 0.4 and 1.0, the agent tends to prefer corridor 14. As the RCS decreases, it then prefers corridors 8, 10, and 2 in that order. Since the RCS is a direct component of the route selection algorithm, it has the largest influence on the selection of a route. This is also a function of the weighting value placed on the RCS value within the route selection algorithm to put a heavy bias toward survivability of the LRS platform.

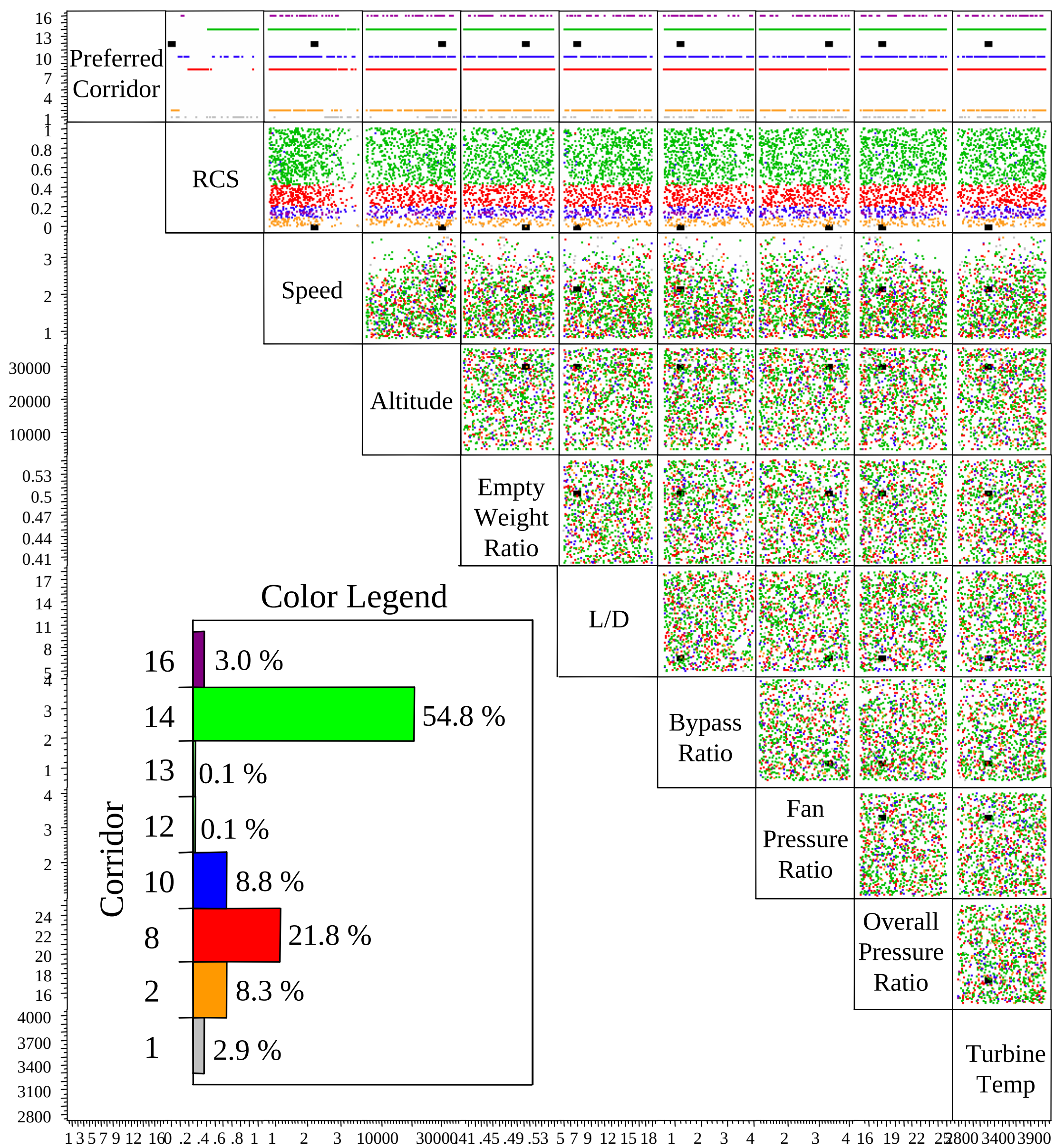


Figure 80: Multivariate Analysis Showing the Relative Sensitivity of Route Selection to Aircraft Technology Variables.

The behavior of the agent as a result of technology infusion can also be observed using the Pareto chart shown in Figure 81. From this figure, it is again clear that RCS is the single greatest contributing factor to an agent's route selection. The remaining parameters that contribute 80% towards the variability of this selection include platform speed, fan pressure ratio, and fan efficiency. It is important to note that the Pareto chart and resulting distribution of corridor selection is based on the ranges of the input distributions to the Monte Carlo simulation used to generate the data. The maximum ranges of the neural networks as defined in Table 18. Lift-to-Drag ratio was varied from 5 to 15, empty weight ratio from 0.4 to 0.55, and RCS from 0 to 1 m^2.

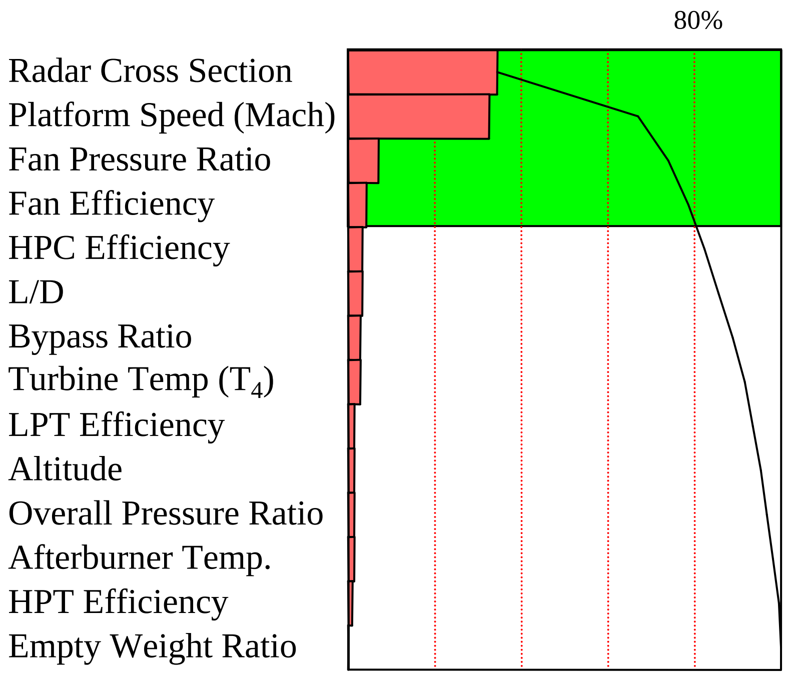


Figure 81: Relative Contribution of Technology Parameters to Route Selection.

This experiment demonstrates that not only does the agent make decisions that differ as the number of SAM sites in a given corridor are reduced over time, but it also changes its decision based on its own physical characteristics. These characteristics are “tuned” by surrogate models inside the intelligent agent. Since the agent seeks the “best” way to accomplish his objectives, tunable cognition models cause the agent to automatically exploit technologies provided to maximize their benefit in the simulation.

5.7 *Step 7: Map Technologies to k-Factors*

According to Hannay and McGinn, "the basic function of technology is the expansion of the realm of practical human possibility" [192]. A simulation can be used to assess any phenomenon for which a model has been created. To demonstrate the effectiveness of the SOCRATES methodology for capability-based technology evaluation, several *representative* candidate technologies and the means to simulate them are discussed in the subsequent sections.

5.7.1 Propulsion and Aerodynamic Technology Advancements

Technology assessment studies have long examined the impact of aerodynamic, structural, and propulsion system improvements in isolation, and only in recent years have advances in modeling and simulation enabled a holistic analysis of technologies related to both engines and airframes with a quantification of uncertainty [284, 296]. The next logical step is the extension of this analysis process to the system-of-systems level to quantify tradeoffs between candidate propulsion and airframe technologies with respect to capability-level metrics. To bound the problem space, this research studies technologies under consideration by the AFRL Vehicles Directorate (VA) and Propulsion Directorate (PR) including [101, 229]:

- High L/D Tailless Aeroconfiguration (VA)
- Efficient Propulsion Installation (VA)
- Full Envelope Weapon Release (VA)
- Light Weight Thermal Structures (VA)
- Efficient Transonic Planform (VA)
- Affordable Multi-Role Structure (VA)
- Durable High Temp Core and Fuel Efficient Turbine Engine (PR)
- Fuel Efficient Expendable Turbine Engine (PR)
- Fuel Efficient Expendable Scramjet (PR)

These technologies primarily involve drag reduction, increases in propulsive efficiency, weapon release and vehicle operation at transonic speeds, weight reduction, higher turbine temperatures, and improved specific fuel consumption. While each of these technologies can be analyzed in isolation, the SOCRATES methodology is used to compare aircraft and propulsion technologies against the same set of assumptions and with respect to the same capability-level metrics.

5.7.2 Sensor Technologies for TCT Attack

The problem of time critical target strike that confounded operations in the Persian Gulf War and Kosovo can be partially addressed by developing new sensor capabilities that include "deep-look, long-dwell, all-weather/day-night operations and acceptable survivability in the face of advanced air defenses" [3]. The AFRL Sensors Directorate is pursuing advanced capabilities and processing algorithms such as Automatic Target Recognition and Sensor Fusion that enable the identification and discrimination of TCTs amidst background clutter [443].

Since models of these high fidelity sensor algorithms are not available for this research, a simple model that uses the radar range equation is used to model the ability of a platform to detect and track targets in its surrounding environment. By modulating the sensor range, technologies related to signal-to-noise reduction, increased sensor power, and better discrimination algorithms can be modeled at the most basic level. Using the SOCRATES methodology and the aforementioned modeling and simulation environment, the benefit of sensor technologies are quantified in relation to vehicle and architecture technologies for LRS. The primary focus of sensor technologies is on the TCT attack mission where onboard sensing becomes a critical driver in target tracking and engagement.

5.7.3 Advanced Standoff Weapons

One tactic for LRS is to employ high speed platforms and weapons to perform the "engage" function of the kill chain in a shorter amount of time. This concept is supported by General David M. Edgington, director of global power programs in the Air Force acquisition office who, along with Air Force Chief of Staff Michael Moseley, advocates a new Long Range

Strike bomber by 2018 “to be able to strike targets in near-real time.” Edgington also noted that the current bomber fleet “lacks survivability, especially during daylight hours” [205]. One potential strategy for LRS is to ignore the development of a costly new platform and pursue advanced standoff weapons that can be employed from existing legacy platforms in the current fleet.

Although the 2006 Congressional Budget Office study on alternatives for long-range ground attack systems did not recommend a specific alternative for LRS, the study highlights an “arsenal aircraft” armed with *supersonic cruise missiles* as a low-risk, low-cost alternative that provides reasonable response time and adequate global coverage [37]. A number of studies have examined cruise missiles of various speeds and ranges for a variety of LRS-like missions [338]. For example, a proposed concept, the Revolutionary Approach to Time-critical Long Range Strike (RATTLRS) is envisioned as a “high-supersonic cruise missile capable of speeds greater than Mach 3 that can be launched from Navy and Air Force platforms including surface ships” [465]. Goals for RATTLRS include high-supersonic speed (Mach 3-4), long range (up to 1000 km), fuel efficiency, and the ability to attack mobile, time-critical, or hardened/buried targets [173].

Subsonic, supersonic, and hypersonic cruise missiles must be compared to air vehicle and C2 technologies to assess the relative contribution of these revolutionary weapons to increasing the speed of the kill chain. While aerodynamic and trajectory models specific to the high-speed flight regime are difficult to develop, a first-order assessment of the viability of these weapons can be conducted by modulating the speed, range, drag coefficient, and fuel efficiency of the munition concepts in question. Using a simplified parametric missile model within FLAMES, a variety of munition architectures can be compared to the suite of air vehicle technologies within the same scenario and using the same assumptions.

5.7.4 Comparison of High Speed Weapons and Area Dominance Munitions

In contrast to the previous approach, a non-intuitive paradigm for munition development uses extremely low speed munitions, pre-positioned near potential targets that loiter and wait for specific target information provided by the battle manager. This concept is referred to as the "Area Dominance Integrating Concept" and involves "continued presence of a constellation of lehal, minature, high endurance, multi-shot, persistent munitions capable of cooperatively striking high priority targets" [339].

The first example of such a system is the marriage of the RQ-1 Predator reconnaissance UAV with the AGM-114 Hellfire missile to create a remotely piloted attack platform. The MQ-1 armed Predator was tested in 2001 and implemented operationally in early 2002. Another example is the Lockheed Martin Low-Cost Autonomous Attack System (LOCAAS), a turbojet powered 100 pound "area dominance" munition that loiters after release for up to 30 minutes at 750 feet to identify and attack targets using a LADAR[25] seeker [261]. Other area dominance munition concepts such as the Low Cost Mini Cruise Missile (LCMCM) and Dominator are under development at the AFRL Munitions Directorate at Eglin, AFB [340, 341]. Examples of some area dominance concepts are shown in Figure 82.

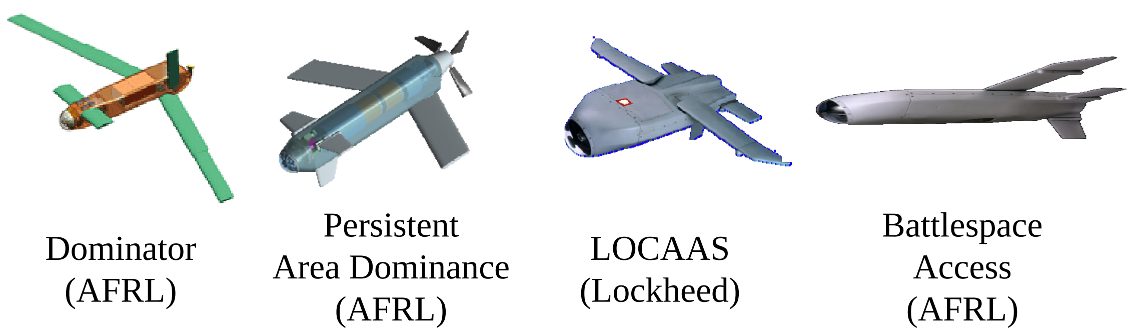


Figure 82: Examples of Some Proposed Area Dominance Munition Concepts [146, 248].

5.7.5 Improving the Targets-per-Sortie Rate

The Gulf War Air Power Survey notes that Operation *Desert Storm* was the first time a single sortie could be assigned to a single target with a high probability of target destruction. In 1999's Operation *Allied Force*, the B-2A bomber "delivered up to 16 GPS-guided GBU-31 joind direct-attack munitions (JDAMs) from 40,000 ft, usually through cloud cover, against hardened enemy targets, including command bunkers and air defense facilities" [254]. In

[25] Often incorrectly referred to as "laser radar," Light Detection and Ranging (LADAR) uses the properties of optical scattering to identify and range distance targets.

September 2003, the B-2A bomber demonstrated the capability of deploying eighty Mk-82 500 lb JDAMs at eighty separate targets [402]. A key architecture technology is the number of engagements or targets per sortie. The model used for ground attack in the simulation allocates one bomb per target per sortie. When multiple munitions are equipped on a single platform, the battle manager can allocate multiple targets to the platform, until its fuel falls below a defined threshold or all munitions are expended.

Increasing the number of munitions per platform enables a study of the effects of multiple-target-per-sortie missions in conjunction with other technology factors described in this section.

5.7.6 Advanced Radar Absorptive Materials

According to Watts, flight at Mach numbers above 2.4 pushes the limits of current radar absorptive materials (RAM) [478]. The Alternate High Frequency material (AHFM) program at the AFRL Materials and Manufacturing Directorate developed a new spray-on coating for the B-2 bomber that allowed retention of stealth characteristics while minimizing maintenance time due to the previously used tape-and-caulk method [442]. One key technology area to be traded with platform and munition attributes is the amount of stealth needed to prosecute missions against a hostile IADS.

5.7.7 Defining a Baseline

The evaluation of technology performance and quantification of technology gaps is difficult without establishment of a baseline against which to compare. As mentioned in Section 2.4.2.3, the B-2A *Spirit* stealth bomber is the closest system to the LRS capability need statement with a primary shortcoming in responsiveness due to its subsonic speed. Since the multi-target attack, high-survivability, large payload, and extended range characteristics of the B-2A serve as a "best-in-class" example of LRS capability, this system is used as the baseline against which technology impact factors are referenced[26]. A reasonable munition baseline is the GBU-31 JDAM, a gravity dropped glide bomb; however, the JASSM is

[26] Stonier gives the RCS of the B-2A as 0.01 m^2 and the F-117A as 0.1 m^2. The F-117 value is used to make the bombers susceptible to notional Iraqi threats

intended as an air-launched medium-range subsonic cruise missile which could also serve as a suitable baseline. A propulsion system of the caliber of the Pratt and Whitney F-119 that powers the F-22A *Raptor* is considered an acceptable engine baseline, although precise values for component efficiencies and turbine temperatures can only be inferred from open source literature. The baseline values defined by these assumptions are defined in Table 20.

Table 20: Baseline Values for Technology Impact Factors.

Factor	**Value**	**Source**
GTOW	154,211 kg (340,000 lbs)	Ref. 467
Empty Weight Ratio	46.5%	Ref. 16
Thrust/Weight Ratio	20.40%	Ref. 467
Payload Weight Ratio	11.70%	Ref. 467
W/S	68 lb/ft^2	Ref. 16
C_D	0.03	Assumed
C_L	1.5	Assumed
TSFC	0.67 lb_f/lb_m-hr	Ref. 277
Radar Cross Section	0.1 m^2	Ref. 386
Platform Speed	Mach 0.85	Ref. 467
Weapon Speed	Mach 0.85	Ref. 16
Weapon Range	24 km (15 miles)	Ref. 16
Weapons	Up to 80 (4 Used)	Ref. 402
Fan Efficiency	91%	Assumed
Turbine Temp (T4)	1778°K (3200°R)	Assumed

5.7.8 Determining Technology Impacts

Kirby advocates the use of a Technology Impact Matrix (TIM) to quantify the maximum and minimum benefits of a technology in terms of technology k-factors that are applied to discipline level metrics [240]. This approach is valid for exploratory forecasting where specific technologies are being analyzed for their impact on a system. While the previous sections highlight some candidate technologies, these are representative examples of the type of trade studies that should be performed to assess system, subsystem, and architecture technologies for a LRS system architecture. A notional TIM is shown in Figure 83. Blue shaded technologies represent platform and subsystem technologies identified by AFRL, yellow shading represents other platform technologies, tan shading highlights munition technologies, and light green shading identifies architecture technologies. The positive and negative factors in this matrix indicate the approximate increase or decrease in the k-factor representing the metric identified and is representative only: ***AFRL documentation and associated literature does not enumerate exact values for the technology impact factors***. They are thus estimated by the author.

Using the TIM as a list of representative technologies and impacts, it is feasible to bound the distribution of k-factors with respect to the maximum positive and negative impacts as shown in Figure 83. These bounds are represented in the range of the design of experiments used to create surrogate models in Steps 8 and 9 of the SOCRATES method. If the surrogate models are appropriately created, the resulting surrogate model-enabled tradeoff environment enables parametric variation of any technology included within the range of the design of experiments. Finally, it is important to note that several of the technologies such as the Loitering Area Dominance Munition require a discrete variation in the system architecture as opposed to modulation of one or more k-factors. This is because such architecture technologies use physics or cognition models to calculate MoEs that differ from the baseline case.

The flexible nature of the simulation environment created for this research allows new technologies to be easily added to the TIM in Figure 83. The subsequent sections show how the example technologies in this matrix can be rapidly analyzed using the parametric tradeoff environment and how the benefits of the technologies can be quantified using modeling and simulation.

#	Technology Description	Platform									Munition			Engine		C2/Sensor	
		k-GTOW	k-Empty Weight Ratio	k-Payload Weight Ratio	k-W/S	k-CD	k-CL	k-TSFC	k-Radar Cross Section	k-Platform Speed	k-Weapon Speed	k-Weapon Range	k-Weapons	k-Fan Efficiency	k-Turbine Temp (T4)	k-C2 Response Time	k-Sensor Range
T1	High L/D Tailless Aeroconfiguration					-10%	+10%										
T2	Efficient Propulsion Installation							-5%						+2%			
T3	Full Envelope Weapon Release									+10%	+10%						
T4	Light Weight Thermal Structures	-20%	-20%		+10%												
T5	Efficient Transonic Planform					-5%				+10%							
T6	Affordable Multi-Role Structure			+10%									+100%				
T7	Durable High Temp. Core and Fuel Efficient Turbofan							-10%							+15%		
T8	Fuel Efficient Expendable Turbine Engine											+500%					
T9	Fuel Efficient Expendable Scramjet										+500%	+500%					
T10	Advanced Radar Absorptive Materials		-10%						-90%								
T11	Long-Range Sensors for TCT Attack																+400%
T12	Long Range Subsonic Cruise Missile											+200%					
T13	Supersonic Cruise Missile										+100%	+200%					
T14	Hypersonic Cruise Missile										+500%	+200%					
T15	Loitering Area Dominance Munition										-50%	-50%	+100%			-50%	
T16	Advanced C2 Processing for Reduced Response Time															-90%	
T17	Multiple Weapons Per Platform			+20%									+400%				
	Maximum Negative Impact	-20%	-20%			-10%		-10%	-90%		-50%	-50%				-90%	
	Maximum Positive Impact			+20%	+10%		+10%			+10%	+500%	+200%	+400%	+2%	+15%		+400%

Figure 83: Example Technology Impact Matrix for Representative LRS Technologies.

5.8 Step 8: Execute Simulation

Using the three simulations identified in Section 5.2.1.6, cases were executed using a series of Pentium IV computers based on the availability of machines with FLAMES runtime licenses. An execution utility based on Tangen's initial investigation was written for the ModelCenter® framework to import each DoE run into the FLAMES database and track the output results [394, 30].

For the hardened, deeply buried target and decapitation strike scenario, a 20,000 case space-filling design was used. The time critical attack mission was executed using a 2,500 case space-filling design for the baseline architecture and a 2,500 case space-filling design for the area dominance munition-based architecture. These selections were made based on the number of variables in each study and the amount of computational resources available. Finally, the three scenarios were integrated together into a holistic scenario that mimics a short campaign in support of GSTF objectives.

In the HDBT strike scenario, the Baghdad National Air Operations Center was assigned leadership, IADS, and C3 signatures of 81, making it the highest priority target in the scenario. A single platform based at the Khamis Mushait airbase in Southern Saudi Arabia was available for the engagement and was allowed to utilize any of the twenty airspace corridors illustrated in Figure 74. The simulation time was constrained to five hours (simulated). A 20,000 case space-filling design based on the DOE shown in Table 21 was used to execute the cases.

The same space-filling design was used to execute the simulation for the decapitation strike scenario. The primary difference between the two scenarios is that the aircraft in the HDBT scenario has 5 hours to complete its mission. In the decapitation strike scenario, the aircraft has only 90 minutes to strike the target in question.

For the TCT attack scenario, the number of variables was reduced to simplify the trade-off study because additional tactical parameters were examined. To further simplify the analysis, only MoEs related to targets killed were tracked. As a result, the IADS elements were removed to speed up the simulation and TSFC was set to zero for all aircraft elements to eliminate the confounding effect of refueling. Targets were located by an airborne sensor

Table 21: DOE Ranges for the HDBT and Decapitation Strike Scenario.

Variable	**Low**	**High**
GTOW	35,000 lbs)	1,200,000 lbs
Empty Weight Ratio	0.4	0.55
Thrust/Weight Ratio	0.35	1.5
Payload Weight (lbs)	1,000	60,000
W/S	20 lb/ft^2	150 lb/ft^2
Drag Coefficient (C_D)	0.01	0.09
Max C_L	1.5	3
P_{Kill}	70%	100%
Radar Cross Section	0.001 m^2	1 m^2
Platform Speed	Mach 0.85	Mach 4
Weapon Speed	Mach 0.85	Mach 6
Fan Pressure Ratio	1.5	3
Overall Pressure Ratio	15	40
Turbine Temp (T4)	1556°K (2800°R)	1889°K (3400°R)
Enemy Radar Power	10	100
Enemy SAM Density	0%	100%

with a variable sensor range and relayed directly to subordinates. In the baseline scenario, the subordinates were fighters loitering in northern Saudi Arabia. Each fighter was armed with up to four air-launched missiles with a range of 185 km (100 nm) and a speed of up to Mach 4. The P_{Kill} of this munition is 70%. Five airborne sensors each control six fighters and follow a pre-defined patrol pattern in regions of Iraq. These patrol regions were pre-selected based on the presence of TBM launchers in the area. TBM launchers fire missiles at targets in nearby Israel according to a pre-determined pattern, and move/hide according to the activity diagram shown in Figure 67. The ranges of the DoE used to analyze the baseline case are highlighted in Table 22.

The TCT attack scenario was also executed using the area dominance munition concept. In this case, four area dominance munitions were mounted onboard each of five airborne sensors and released at the time of target detection. The airborne sensors fly the same search patterns as in the baseline case. The area dominance munitions fly out to the target area, attempt to locate the target, and release a submunition at the target. The P_{Kill} of each submunition is 70%. Due to their small size and long loiter time, the area dominance munitions necessarily have a much lower speed range than the fighter aircraft in the baseline case. An identical pattern of SCUD launchers and launch timings was used for the area dominance and baseline cases. In addition to the parametric variation shown in Table 22, several tactical variations for the area dominance case were also examined:

- Doubling the number of AD munitions
- Increasing the dispersion of the AD loiter pattern
- Doubling the number of munitions and increasing the dispersion
- Tripling the number of munitions and increasing the dispersion

The results of each of these cases are examined in detail in a subsequent section.

For the GSTF attack scenario, six LRS platforms were used to strike targets over a period of three days (three eight-hour segments). The run time for these simulation cases was between five and twenty minutes, depending on the input parameters used. As a result, a 20,000 case DOE was deemed to large to be executed in a reasonable time frame. A smaller DOE consisting of 2,000 case space-filling design was executed using three computers over approximately five days. To increase the likelihood of generating surrogate models with adequate fits, a smaller design of experiments (both in terms of number of variables and their ranges) with ranges shown in Table 24 was used. For these runs, P_{Kill} of the friendly weapon was set to 100% and the TSFC was not decomposed into its lower-level elements. The results of this set of runs is described in the next section.

Table 22: DOE Ranges for the TCT Attack Scenario (Baseline Case).

Variable	**Low**	**High**
Platform Speed	Mach 0.55	Mach 5.45
Submunitions	1	4
Target Speed (mph)	15	60
Target RCS (m^2)	0.01	100
Target RCS (m^2)	0.01	100
Sensor Range, Locator (km)	2	100
Sensor Range, Attacker (km)	2	100

Table 23: DOE Ranges for the TCT Attack Scenario (Area Dominance Munitions).

Variable	**Low**	**High**
Platform Speed	Mach 0.45	Mach 1.1
Munitions	1	4
Target Speed (mph)	15	60
Target RCS (m^2)	0.01	100
Target RCS (m^2)	0.01	100
Sensor Range, Locator (km)	2	100
Sensor Range, Attacker (km)	2	100

Table 24: DOE Ranges for the GSTF Attack Three Day Scenario.

Variable	**Low**	**High**
GTOW	100,000 lbs)	800,000 lbs
Empty Weight Ratio	0.4	0.55
Thrust/Weight Ratio	0.35	0.7
Payload Weight (lbs)	5,000	30,000
W/S	80 lb/ft^2	120 lb/ft^2
Drag Coefficient (C_D)	0.01	0.09
Max C_L	1.5	3
Radar Cross Section	0.001 m^2	1 m^2
TSFC (lb_m/lb_f-hr)	0.2	0.9
Platform Speed	Mach 0.85	Mach 4
Weapon Speed	Mach 0.85	Mach 6
Weapon Range	24 km (13 nm)	75 km (41 nm)
Enemy Radar Power	10	100
Enemy SAM Density	0%	100%

5.9 Step 9: Generate Surrogate Models

As previously mentioned, surrogate models not only enable intelligence for agents within the simulation framework, but can also be used to enable rapid trade studies on the output results. The difference between these two uses is that in the first case, surrogate models are generated *a priori*, inserted inside the simulation, and exercised at runtime to provide information to intelligent agents. On the other hand, surrogate models used for design space exploration are created around the output data of the entire simulation after the results are generated.

Since thousands of cases are executed to generate surrogate models with excellent coverage of the design space, a key question that arises is "why not just use this data and skip the surrogation process entirely?" Certainly, this data is the most accurate available since the output data has been run directly through the simulation code. The only source of error in this data is from the models used to generate the data; however, this is the *only* data available, and trade studies can only be performed using the values in the data table. In contrast, a correctly constructed surrogate model has predictive capabilities within the ranges of the design of experiments that allows generation of additional "evidence" for making decisions.

Under the current circumstances, it may not be possible to generate surrogate models due to the number of cases required and the run time needed to generate one case; however, this does not invalidate a generalized methodology. In early 2002, a bank of ten Sun workstations was utilized 24 hours a day for a week to generate 1,024 cases for a NASA technology study [283]. In contrast, by 2006 the same amount of data could be generated using a single Windows-based platform in less than eight hours. In some cases, analysis data is used in the absence of surrogates and future work with the Air Force will leverage Grid computing to generate larger data sets for regression.

5.9.1 The HDBT and Decapitation Strike Scenarios

The results of the HDBT and decapitation strike scenarios were analyzed using the BRAINN tool. The goodness of fit metrics for the four neural network equations (targets killed and platforms lost for each of the two situations) are shown in Table 25.

Table 25: Goodness of Fit Statistics for HDBT and Decapitation Strike Predictive Neural Networks.

	HDBT Strike		Decapitation Strike	
Parameter	**Targets Killed**	**Platforms Lost**	**Targets Killed**	**Platforms Lost**
Validation Data %	25	25	25	25
Test Data %	5	5	5	5
Training Time (s)	3600	3600	3600	3600
Iterations	2	2	2	2
Hidden Nodes (Low)	6	6	6	6
Hidden Nodes (High)	14	14	14	14
Training % Correct	99.6143	98.7787	98.7571	99.9429
Validation % Correct	98.76	97.52	96.78	99.64
Test % Correct	98.8	98	96.9	99.7
Optimum Nodes	6	8	10	8
Number of Points	20,000	20,000	20,000	20,000

The error in these two cases is quite low. The neural network incorrectly predicts the results of only 366 out of 20,000 cases for the decapitation strike case and only 128 out of 20,000 cases for the HDBT strike case. Nevertheless, not all scenarios have results this "well-behaved." A technique is needed to identify the sources of error and prioritize the execution of additional cases to increase the resolution of the design in this region. For each case in which the simulation has been run, both an actual and predicted value for the platforms lost and targets killed equation are available. The 494 cases where the neural network fails at the predicted response can be analyzed as a subset of the data. The multivariate analysis tool in JMP®, usually used for analyzing correlations between variables or performing the inverse design technique, can also be used to identify what groups of cases have in common. In this case, the multivariate plot is used in Figure 84 and Figure 85 to identify regions where points are clustered. Histograms are shown at the intersection of each design variable, and are generally uniformly distributed. A uniform distribution is an indication that the error in the predicted response is not a function of the variable in question. On the other hand, when munition range is near the minimum value in the DOE, error is disproportionately present. These points are highlighted in red

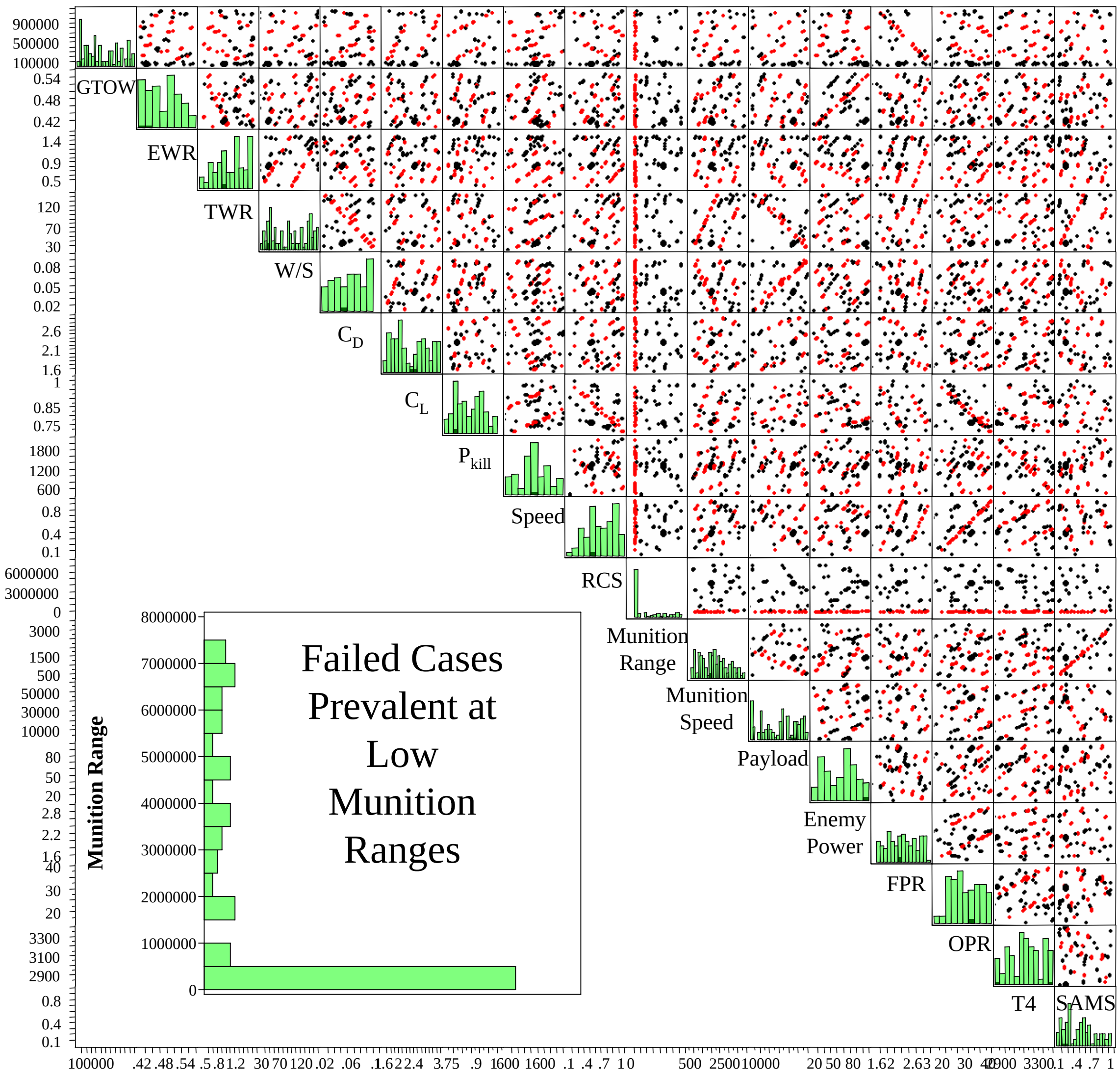


Figure 84: Analysis of Error for the HDBT Strike Case (128 Incorrect Predictions)

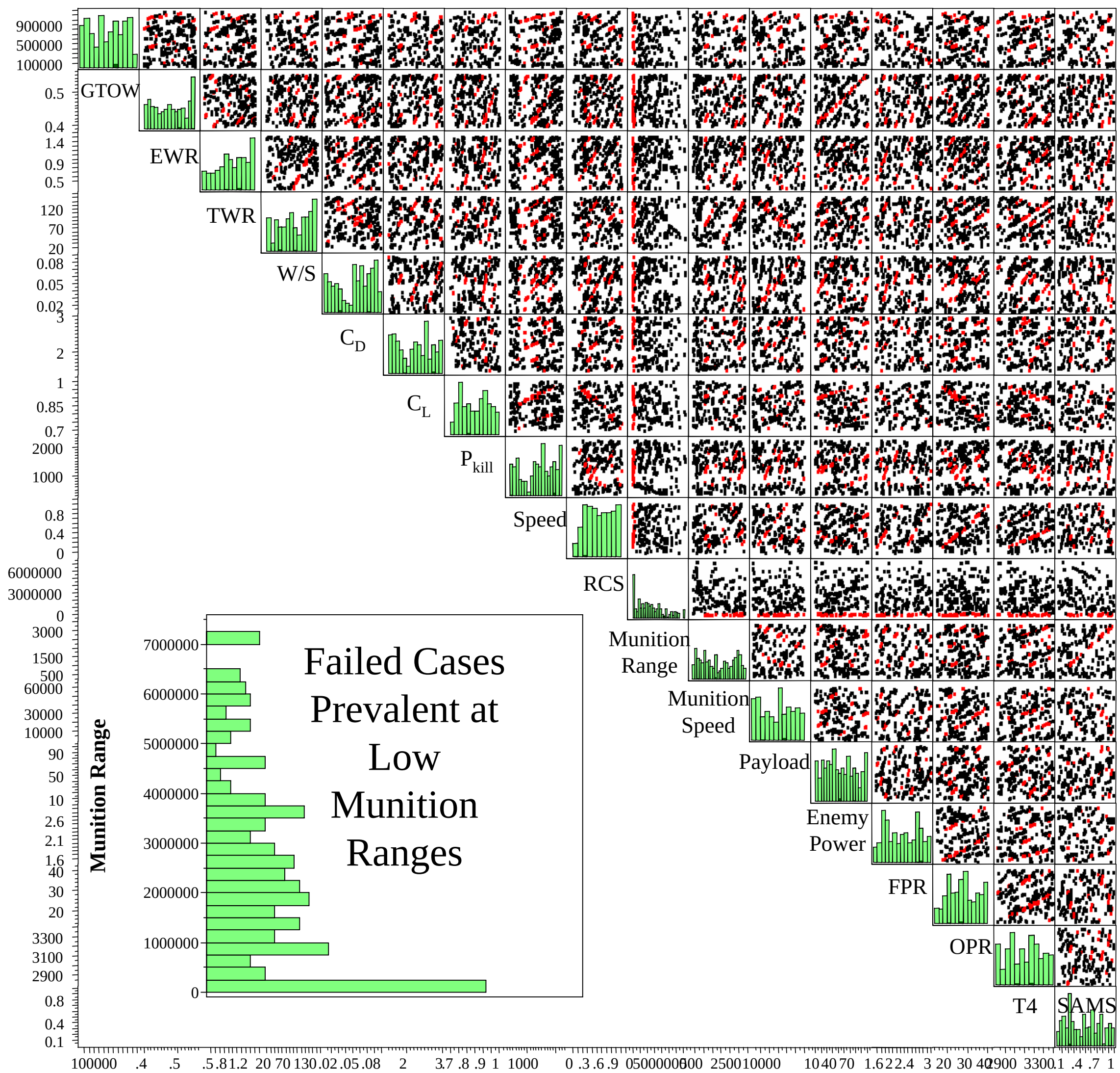


Figure 85: Analysis of Error for the Decapitation Strike Case (366 Incorrect Predictions)

for emphasis. This means that either there is a problem with the model at low munition ranges or more cases must be run at these treatment settings to provide the neural network with enough data to make a valid prediction. In this case, it was determined that a fast platform cannot release its weapon and have it fall to the target along a ballistic trajectory at weapon ranges below a certain threshold. The ranges were set to provide "bomb-like" and "long range cruise missile-like" cases for the HDBT and decapitation strike; however, the resolution at the lower end of the design space is poor because the cases are uniformly distributed from small to very long ranges.

This example demonstrates a technique for ascertaining the source of error in a complex multidimensional problem using a graphical visualization technique in combination with statistical analysis, and is used in the two subsequent scenarios to identify areas where additional runs are needed.

5.9.2 The Time Critical Strike Scenario

The results of the output DOEs from the TCT strike scenarios were regressed using the BRAINN tool. Because the number of variables was much smaller than in the other scenarios, the BRAINN tool typically reached convergence in under one minute, as opposed to the hour-long runs executed for the more complex scenarios. The goodness of fit metrics for the TCT strike baseline scenario using fighters located in Northern Saudi Arabia is shown in Table 26. The results from the area dominance case are shown in Table 27.

While the R^2 values for the baseline case are reasonably high, the goodness of fit metrics for the area dominance case are substantially lower. This may be due to some of the other confounding factors in the area dominance scenario related to how and when the TBM launchers are detected and tracked. Another major difference between the two scenarios is the fact that the starting location for the baseline case is almost always the same, while the starting location for the area dominance case is unknown. Despite numerous attempts to improve the fit of the equation or trace the cause of the confounding factors, the goodness of fit metrics did not improve significantly. The trade studies performed for this scenario uses the actual data from the baseline and variation runs as described in Section 5.8.

Table 26: Goodness of Fit Statistics for Time Critical Strike Baseline Scenario.

Parameter	**TBM % Killed**	**TBM Killed**	**Missiles Fired**	**TBMs Fired**
Validation Data %	15	15	15	15
Test Data %	6.25	6.25	6.25	6.25
Training Time (s)	3600	3600	3600	3600
Iterations	10	10	10	10
Hidden Nodes (Low)	5	5	5	5
Hidden Nodes (High)	25	25	25	25
Training R^2	96.33	96.19	96.17	96.37
Validation R^2	93.20	93.16	93.70	93.30
Test R^2	96.38	96.07	95.09	96.31
Optimum Nodes	21	17	19	17
Number of Points	3,200	3,200	3,200	3,200

Table 27: Goodness of Fit Statistics for Time Critical Strike Area Dominance Scenario.

Parameter	**TBM % Killed**	**TBM Killed**	**Missiles Fired**	**TBMs Fired**
Validation Data %	15	15	15	15
Test Data %	6.25	6.25	6.25	6.25
Training Time (s)	3600	3600	3600	3600
Iterations	10	10	10	10
Hidden Nodes (Low)	5	5	5	5
Hidden Nodes (High)	25	25	25	25
Training R^2	83.27	83.24	93.88	83.05
Validation R^2	84.34	84.61	94.68	83.15
Test R^2	80.76	82.01	93.28	80.53
Optimum Nodes	11	9	15	9
Number of Points	3,200	3,200	3,200	3,200

5.9.3 The GSTF Attack Three Day Scenario

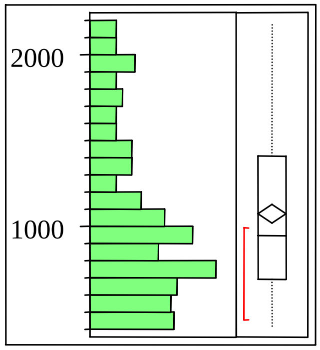
As described in Section 5.8, originally a 2,000 case space-filling DOE was used to execute the simulation runs for the GSTF attack three-day scenario, and 1,000 random cases were used to validate the neural network equations. The number of cases was chosen based on the amount of time it took to generate a single case and the level of resources available to execute cases. The goodness of fit metrics for the targets killed (continuous) and platforms lost (discrete) equations are shown in Table 28.

The goodness of fit statistics for the predictive neural network equation are shown in Figure 87. The model fit error (see Section 5.5.5) is indicative of the ability of the model to predict the values of the 2,000 cases used to generate the model. The model representation error describes the ability of the model to predict the values of the additional 1,000 random cases that were not used to create the model. The second error measure exceeds +/- 23% for the targets killed response and 12% for the platforms lost response.

An analysis of the error in the previous case using the graphical technique described in Section 5.9.1 indicated that a majority of the errors in the GSTF attack case occurred at high SAM densities: there are simply not enough treatments at this level for the neural network to be able to develop accurate predictions. An further analysis of the 333 outlier cases from the first training pass is shown in Figure 86. Based on the observation that cases at low platform speed and TSFC have a high degree of error, an additional 500 cases were executed with a speed between Mach 0.85 and 1.6 and a TSFC from 0.2 to 0.4. An additional 2,000 random cases were executed with SAM densities ranging from 0.8 to 1.0 to provide more treatments in a high threat environment. Of the 5,000 cases executed, 4,625 were valid (3,854 regular, 771 validation).

Another observation made from the data is that while a quantitative assessment of the number of platforms lost is desired, since there are so few results at treatment levels between zero and six, the fit of the neural network equation at these levels is poor. When the response is reduced to a boolean that is unity when any platforms are lost, the predictive capability of the neural network equation improves dramatically: the error of the boolean response is only 0.2% (10/4265).

Platform Speed



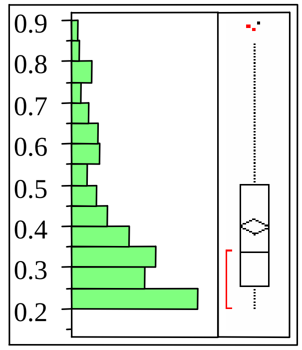
Quantiles

100.0%	maximum	2181.1
99.5%		2177.9
97.5%		2109.8
90.0%		1898.6
75.0%	quartile	1406.2
50.0%	median	947.5
25.0%	quartile	691.6
10.0%		515.2
2.5%		433.5
0.5%		411.1
0.0%	minimum	408.1

Moments

Mean	1071.2472
Std Dev	489.57116
Std Err Mean	26.828334
upper 95% Mean	1124.0222
lower 95% Mean	1018.4723
N	333

Platform TSFC



Quantiles

100.0%	maximum	0.89326
99.5%		0.88731
97.5%		0.81876
90.0%		0.67806
75.0%	quartile	0.50399
50.0%	median	0.33961
25.0%	quartile	0.25752
10.0%		0.21626
2.5%		0.20587
0.5%		0.20110
0.0%	minimum	0.20035

Moments

Mean	0.3989695
Std Dev	0.1771727
Std Err Mean	0.009709
upper 95% Mean	0.4180684
lower 95% Mean	0.3798706
N	333

Figure 86: Analysis of 333 Outlier Cases Contributing to High Predictive Error.

Using the resulting 4,625 cases, a second set of neural networks was created. The goodness of fit metrics for these equations are shown in Table 29. The model fit error and model representation error for the second iteration are shown in Figure 88. After re-executing additional cases in the regions with high numbers of failed cases, the error has been reduced to within an acceptable range for technology forecasting.

Table 28: Results from GSTF Attack Three-Day Scenario Predictive Neural Network Training.

	Response	
Parameter	**Platforms Lost**	**Targets Killed**
Validation Data (#)	750	750
Test Data (#)	250	250
Training Time (s)	3600	3600
Hidden Nodes (low)	6	6
Hidden Nodes (high)	16	16
Iterations at Each	4	5
Training % Correct or R^2	92.4	94.8298
Validation % Correct or R^2	82.4	92.5264
Test % Correct or R^2	86	91.245
Optimal Nodes	6	14
Number of Cases	3,000	3,000

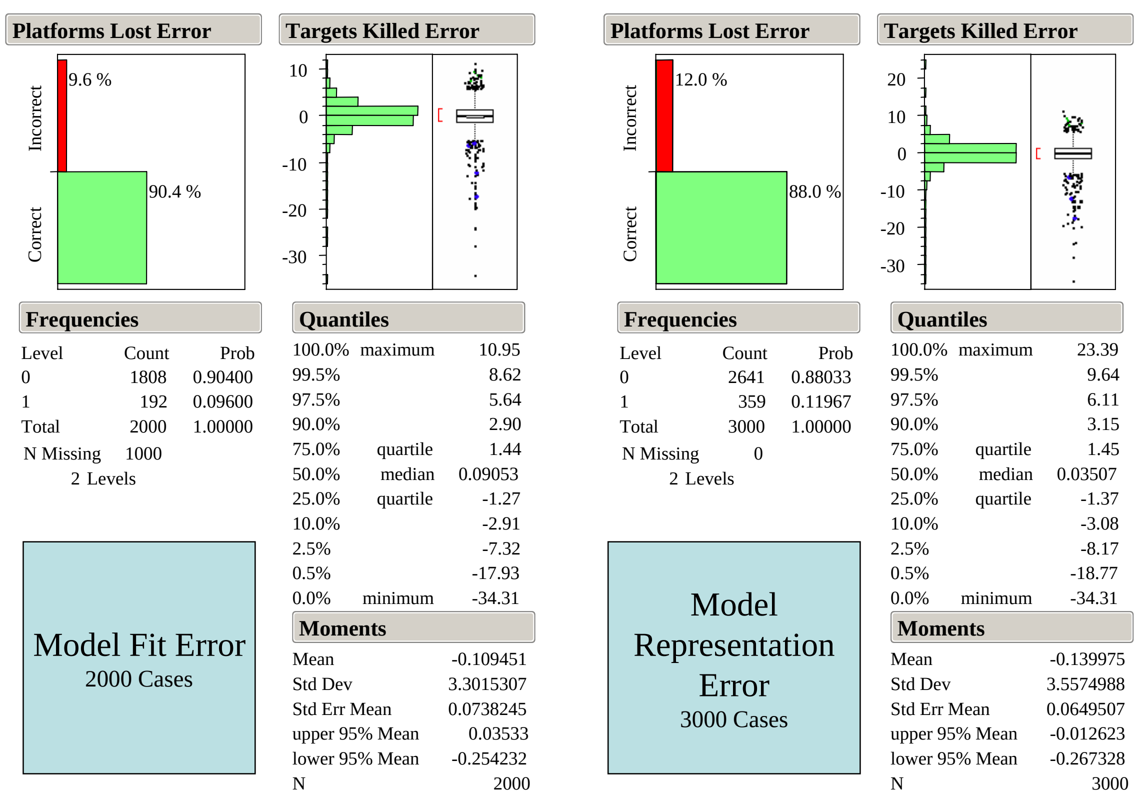


Figure 87: Model Fit Error and Model Representation Error for the GSTF Attack Three-Day Scenario Predictive Neural Network Training.)

Table 29: Results from GSTF Attack Three-Day Scenario Predictive Neural Network Training (Round 2).

	Response		
Parameter	**Platforms Lost**	**Lost (Boolean)**	**Targets Killed**
Validation Data (#)	750	750	750
Test Data (#)	250	250	250
Training Time (s)	3600	3600	3600
Hidden Nodes (low)	5	5	6
Hidden Nodes (high)	11	15	16
Iterations at Each	4	4	5
Training % Correct or R^2	91.8197	100	97.3937
Validation % Correct or R^2	85.5714	98.5714	96.1215
Test % Correct or R^2	87.7828	99.5475	95.9102
Optimal Nodes	5	10	14
Number of Cases	4,625	4,625	4,625

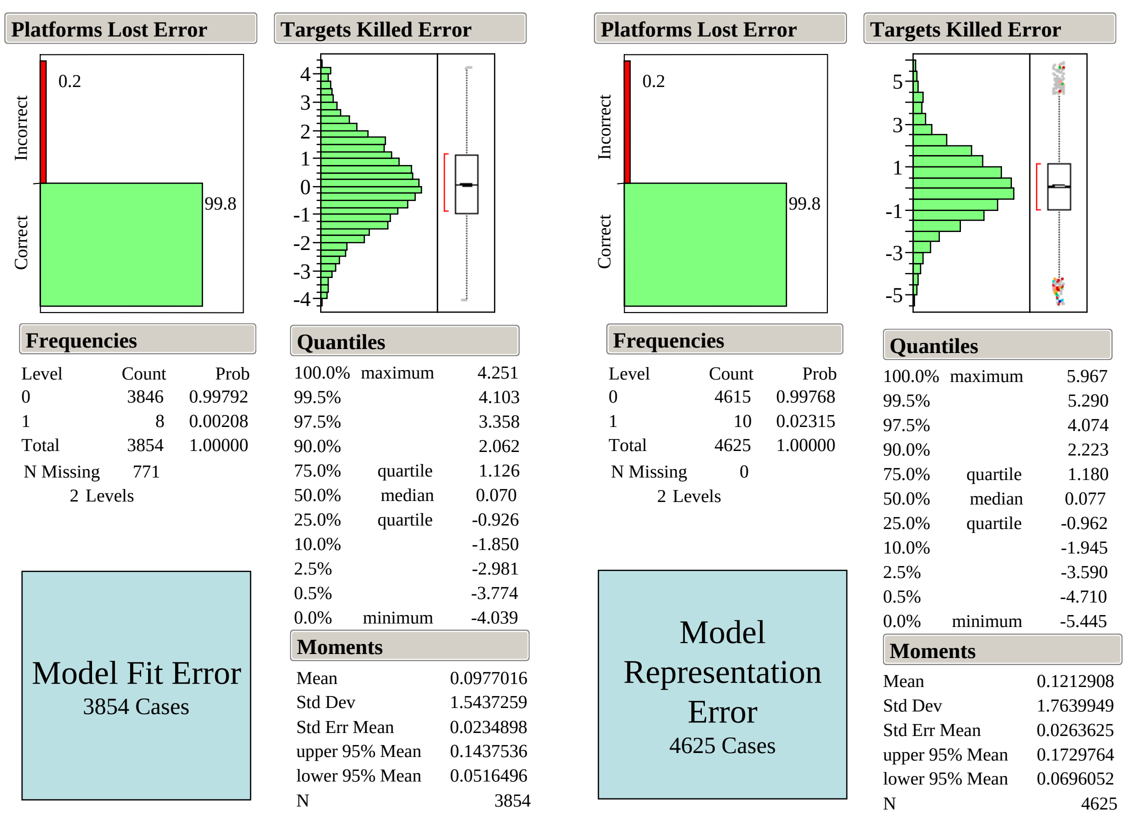


Figure 88: Model Fit Error and Model Representation Error for the GSTF Attack Three-Day Scenario Predictive Neural Network Training (Round 2).

5.10 Step 10: Perform Trade Studies and Evaluate Technologies

"What this task requires in the way of higher intellectual gifts is a sense of unity and a power of judgement, raised to a marvelous pitch of vision, which easily grasps and dismisses a thousand remote possibilities an ordinary mind would labor to identify, and wear itself out in doing so."

-Carl Von Clausewitz,

On War [100]

According to Kass, "the characteristic that separates experimentation from all other research methods, is manipulating something to see what happens" [233]. The first nine steps of the SOCRATES method created a "virtual laboratory" that is a parametric, surrogate-model enabled, hierarchical tradeoff environment. The step that remains is to use the environment to understand the impact of technologies on top-level capabilities and facilitate reasoned decision making. Although the surrogate model encapsulates some degrees of freedom related to the operational assumptions and scenarios employed, the "dials" available to the user still allow near-infinite variation of the properties of the system-of-systems.

First, the heart of the SOCRATES method is the ability to perform quantitative analysis across hierarchical levels of the system of systems. Using the inverse design technique developed by Mavris, Biltgen, and Ender, capability goals at the top level can be related to system attributes and technology characteristics at the system and subsystem level [62]. Using this technique, tradeoff studies and "what-if" games can be performed dynamically and parametrically to identify regions of interest for technology development.

Also, surrogate models enable the analysis of data in bulk: once they are created, the time it takes to generate a single result becomes trivial. As mentioned in Section C.11, probabilistic techniques can be used to quantify uncertainty in estimates and assess the probability of meeting targets across multiple MoEs.

Finally, techniques and methods for visualizing data without creating a "sensory overload" are needed. Multi-dimensional information must be displayed in a manner that enables decision making and does not confound identification of a solution with too many degrees of freedom.

5.10.1 Visualizing the Results

According to the National Academy of Sciences, a major shortcoming of analysis of simulation-based data is that "commercial off-the-shelf visualization techniques are not yet available for high-dimensional data and dynamic data" [310].

The large amount of data generated in the analysis of military simulations is difficult to interpret without advanced visualization capabilities. Mavris has recognized JMP® by the SAS Institute as a leader in statistical analysis and visualization. Graphics are central to statistical analysis in every platform inside JMP®. Some of these platforms include "the prediction profiler, which shows response surface slices through each factor; the contour profiler, which shows horizontal response surface slices with respect to two factors at a time, and the surface profiler, which shows a 3-D rendered surface" [109].

The prediction profiler, summarized in Section C.10.2, is useful for viewing the partial derivatives across the opportunity space and interpreting interactions between variables. The contour profiler extends this one-dimensional analysis into two dimensions. Its primary function is constraint analysis and requirement satisfaction, as shown in Figure 115. A Pareto chart, described in Section C.10.1, is useful for determining the causal relationships of the variability of a response under certain conditions. JMP® also features distribution analysis using histograms. When discrete values are explored, a "mosaic plot" can also be created. This feature is also useful in determining the cause of observed behaviors and is often used to trace the impact on a response back to one or more dominant factors. Typically, these factors can then be fixed at a given value to explore the variability due to other factors while certain values are held constant.

Throughout this research, the JMP® multivariate analysis tool used to enable inverse design and capability discovery by visually displaying the relationships between multiple variables. While this tool is generally used to visualize the correlation between several variables, when used as described in Section C.4, it provides a unique capability to visualize trends and regions of interest. When coupled with surrogate models and the built-in Monte Carlo simulator, the JMP® multivariate analysis is used to quickly explore the design space as degrees of freedom are locked and unlocked. It is important to note that this is not a tool for the novice, but for the expert. Initially, the multivariate analysis tool can be confusing because of the richness of the data it provides. A user's initial apprehension is often overcome when the benefits of quickly manipulating inputs and outputs and querying data in multiple dimensions are realized.

In the subsequent sections, the aforementioned visualization platforms is used where appropriate, noting that the choice of visualization platform must be tuned to the analysis question being answered and the type of results desired: there is no universal plot that depicts all the information to answer every question[27].

5.10.2 Establishment of a Capability Baseline

The primary focus of the SOCRATES methodology is to enable quantitative evaluation of how well a candidate system architecture and a portfolio of technologies provide a capability. According to the JCIDS manual, Measures of Effectiveness (MoEs) are the primary means to evaluate how well a current or proposed solution provides capabilities to accomplish tasks:

> *"Use of the MoEs for the assessment is a key component in determining the existence of a [capability] gap and evaluation of proposed solutions... when possible, use the MoEs integrated with other forms of assessment such as modeling and simulation, high resolution planning, wargaming, etc., to develop a clearer picture of the [capability] gap, its significant factors and its relative importance"* [461].

[27] As a disclaimer: the dynamic visualization views developed for this dissertation are difficult to depict in static form. Where possible, multiple angles and settings are included. In general, the settings are dialed to values that tell an interesting story in a two-dimensional, immovable form.

The MoEs defined in Section 5.2.2 are therefore the *capability-level metrics* against which a solution is measured. The difference between desired thresholds in MoEs and the MoE values calculated using current best-in-class solutions defines the multidimensional *capability gap* for each capability. If no gap exists across all MoEs, then current solutions fulfill the warfighter's needs and no new procurement is necessary[28].

When proposing a new system or technology, the cost of the proposed solution must be compared against potential benefit in these capability-level metrics to assess whether the decision to proceed is a good one from the warfighter's perspective. To perform this comparison, it is necessary to establish baseline values for current capability against which proposed solutions can be measured. Since the baseline must be consistent in terms of assumptions and scenario(s) with the technology-infused solutions to be compared, a classified "official" statement of current capabilities is of little use: the baseline must be calculated using the modeling and simulation environment constructed for the evaluation of proposed solutions and validated against real world systems where available. This process is relatively straightforward after a parametric modeling and simulation environment has been created: the user simply "dials in" the settings that represent present-day systems, executes the simulation, and records the MoEs.

The baseline was established by executing the GSTF three day attack scenario using the parameters of the B-2A *Spirit* as summarized in Table 20. Six B-2A bombers were used. A screenshot of the FLASH playback file is shown in Figure 89.

To eliminate the dependance of the scenario results on the munition, the probability of kill for the weapon is set to 100%. If the platform reaches the target and releases the munition, it is destroyed. Since the bombers do not attack TBMs, the hostile missile launchers fire 74 SCUDs into allied territories and no SCUD launchers are killed. An increase in technology is needed to improve this MoE as the B-2 bomber and many of its contemporaries have no means to engage time critical targets.

[28]Due to the constant evolution of threats to national security, this situation does not usually persist for very long.

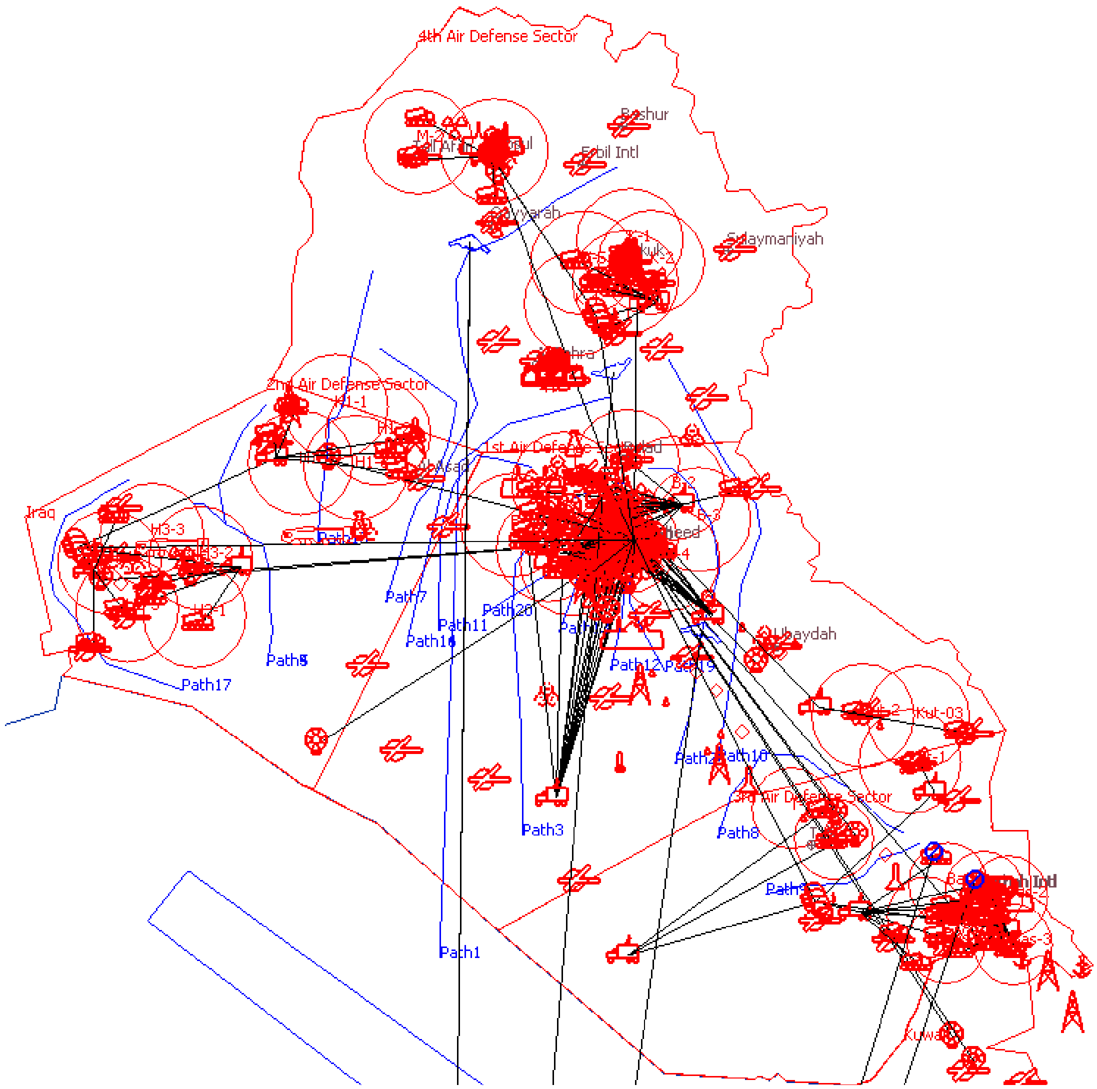


Figure 89: FLASH Playback File of the Baseline Scenario.

Only two of the six B-2 bombers survive the first attack wave and all six bombers are eliminated within eight hours of the scenario start and twenty targets are killed. These results serve as a baseline of the capabilities of present day systems against the modeled threat.

5.10.3 Bottom-Up Analysis of Some Candidate Technologies

Step 7 of the SOCRATES methodology identified a series of notional technologies or technology areas for exploration (see Section 5.7). The enumeration of these technologies not only helps define the models that must be created, but also establishes several example technologies that can be analyzed in a bottom-up manner to calculate their effectiveness. The bottom up, or exploratory forecasting technique is described by Frick as a "push of opportunities" [158].

A complete analysis of candidate platform and munition technologies (T1-T10, T12-T14, and T17) for the HDBT and Decapitation strike missions would require 32,768 runs to only analyze these technology impacts. When incompatibilities are excluded, a daunting 10,240 combinations remain. Using surrogate models created around the HDBT and decapitation strike cases, the execution of this number of runs is trivial: it is nearly instantaneous.

The purpose of the bottom-up analysis is to identify technologies that contribute to a successful mission, that is, where the target is eliminated and the platform survives. A Pareto chart that illustrates the relative contribution of each technology on successful engagements is shown in Figure 90.

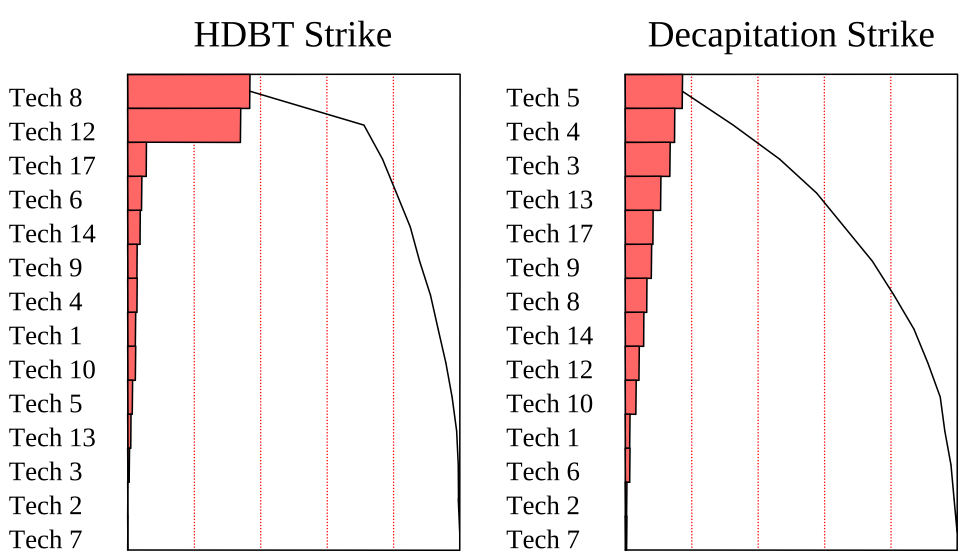


Figure 90: Pareto Chart Illustrating the Relative Contribution of Fourteen Technologies to the HDBT and Decapitation Strike Missions.

For the HDBT strike, Technologies 8 and 12 (Fuel Efficient Expendable Turbine Engine and Long Range Subsonic Cruise Missile) appear to have the greatest influence on the variability of the response. This is reasonable since time is not a factor in the HDBT mission, but increased survivability offered by standoff weapons is desirable. In contrast, the technologies that contribute to the decapitation strike mission shown on the right of Figure 90 are more evenly distributed across platform and munition technologies.

The prediction profiler can be used to ascertain the relative contribution of each technology with respect to the relative impact of all other technologies. An example of the prediction profiler for the HDBT strike is shown in Figure 91 and for the decapitation strike in Figure 92. This figure generally supports the results shown in Figure 90; however, it provides additional information. When the x-axis values are identified as nominal parameters, the prediction profiler functions as a "one-variable-at-a-time" calculator. By moving the hairline from one setting to another, the impact of activating a technology can be calculated using the surrogate models in real time. In this manner, a manual optimization of the technology profile can be conducted with respect to capability-level MoEs; however, this example is fairly limited since only one MoE is included in the tradeoff environment.

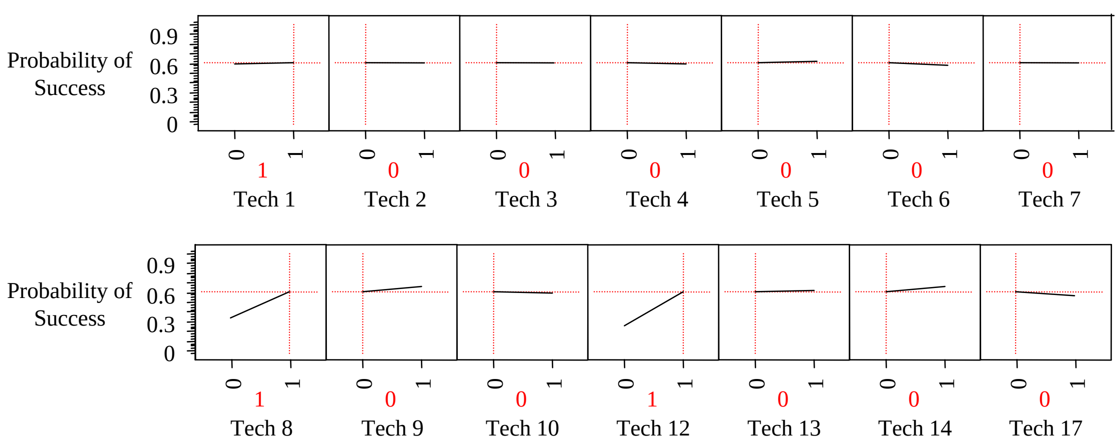


Figure 91: Technology Evaluation Prediction Profiler for the HDBT Strike Mission.

In Figure 91, it again appears that only Technologies 8 and 12 contribute significantly to the overall success of the mission, while there are a number of technology combinations in Figure 92 that can be tuned to maximize the response.

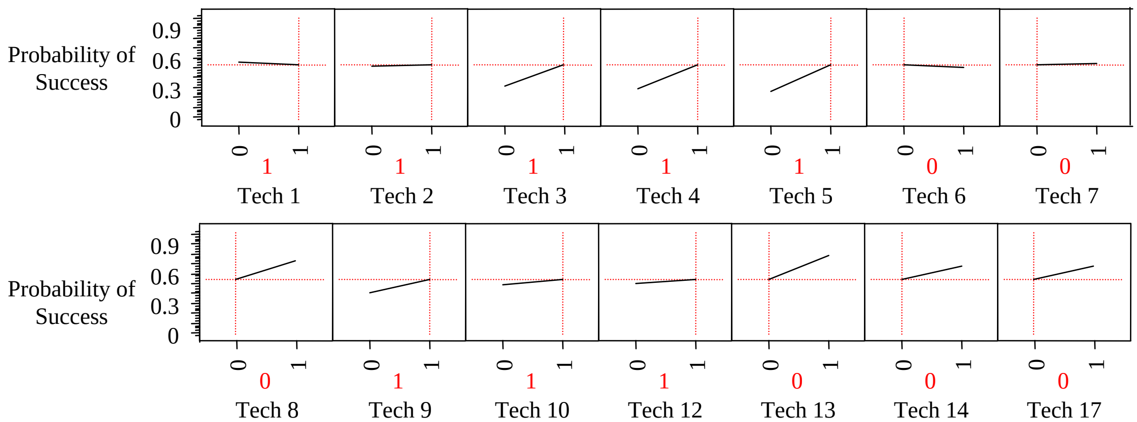


Figure 92: Technology Evaluation Prediction Profiler for the Decapitation Strike Mission.

The next question that evolves from this analysis is "what is the minimum number of technologies needed to ensure a successful engagement for each of the two scenarios?" Recall that in the constructive simulation, the evaluation of effectiveness is a function of the threat and other scenario assumptions. As a result, this question is further bounded by the caveat "against a fixed threat."

The distribution analysis histograms and mosaic plots shown Figures 93 and 94 for the HDBT and decapitation strikes respectively illustrate the distribution of the number of technologies for successful engagements. In general, the HDBT strike requires between five and seven technologies while the decapitation strike requires between six and eight technologies to be successful. It is also interesting to note that there is no single technology which improves the decapitation strike baseline to enable a successful engagement: the minimum number of technologies needed is two. The maximum number of allowable technologies is eleven, bounded by the fact that some technologies are incompatible.

A histogram can again be used to further decompose these results to understand when certain technologies are preferred. The concept of technology portfolios, bundles, or thresholds is useful for this analysis. Risk is generally reduced by implementing a very small number of new technologies in the release of a new system and reserving revolutionary advancements for those technology areas where breakthroughs contribute most significantly to system effectiveness. The analysis previously used to generate the histograms in Figures

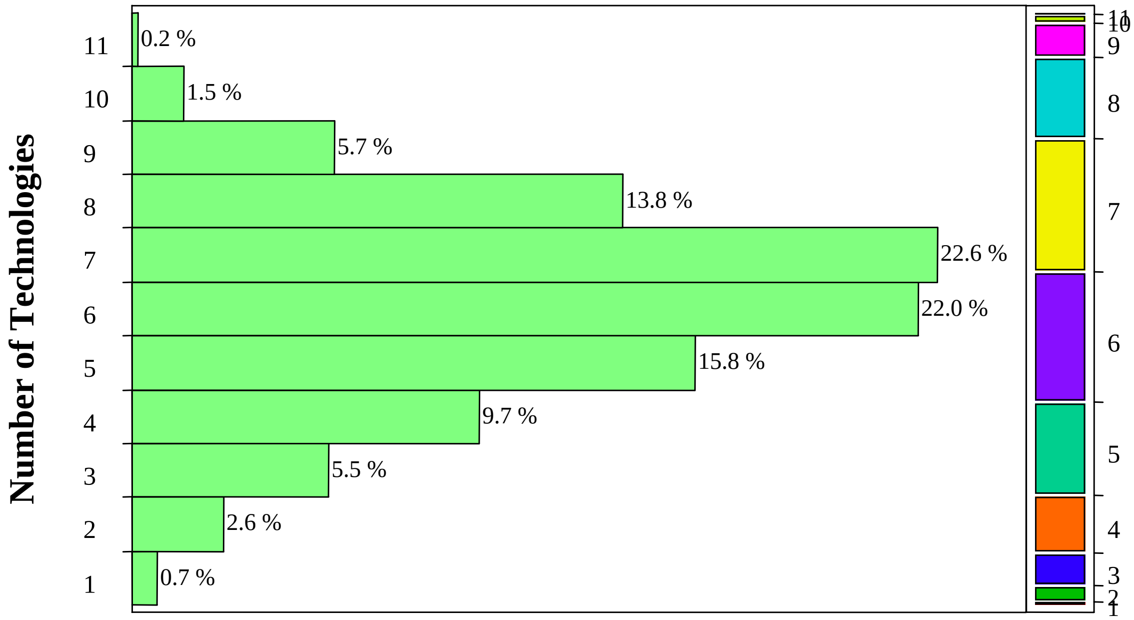


Figure 93: Analysis of the Number of Technologies in a Successful HDBT Strike Portfolio.

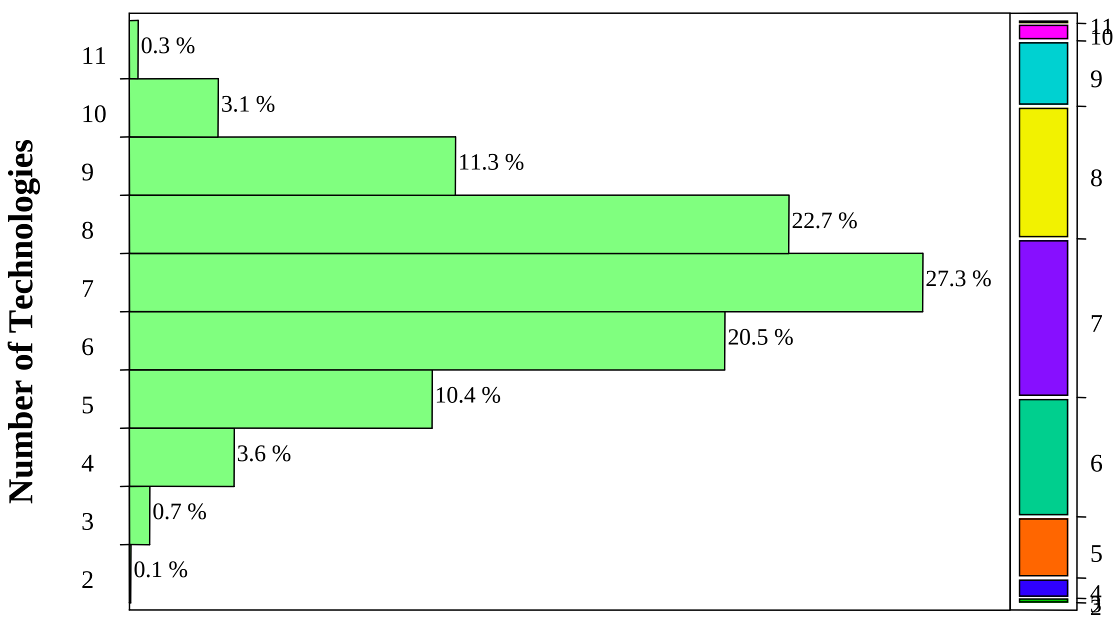


Figure 94: Analysis of the Number of Technologies in a Successful Decapitation Strike Portfolio.

93 and 94 can be used to differentiate technologies into "bundles" as shown in Figure 95 for the HDBT strike and Figure 96 for the decapitation strike. The length of the bars in these figures is indicative of how often a given technology tends to appear in a technology bundle comprised of n technologies, where n is a number from one to eleven. For example, in the HDBT case, there are four technologies that can produce a successful engagement without the addition of any other technologies. Also, there are no cases where Technology 9 and Technology 14 appear in a technology bundle for the HDBT strike: hypersonic cruise missiles do not contribute to effectiveness when time is not a critical mission element. When the frequency of occurrence approaches approximately 9%, this tends to say that the technology is equally likely to be present in a portfolio for the HDBT mission.

Although these two figures are cumbersome, they essentially synthesize information from the previous histogram and the Pareto chart: although Technologies 8 and 12 contribute most significantly to the effectiveness of an engagement, only Technologies 1, 2, 5, and 7 can produce a successful engagement in isolation. These technologies tend to be more likely to be included in the portfolio for a smaller portfolio size; however, as the portfolio size grows, Technologies 8 and 12 tend to be included.

This behavior is also highlighted in Figure 96 for the decapitation strike. The histograms with peaks on the left tend to indicate technologies preferred for smaller portfolio sizes while histograms with peaks to the right indicate that these technologies are favored as portfolio size increases. For the decapitation strike scenario, no single technology can ensure a successful mission; however, there is one two-technology portfolio (Technology 3 and 5). In contrast to the HDBT case, the hypersonic missile technologies have a benefit when time is a factor in calculating effectiveness.

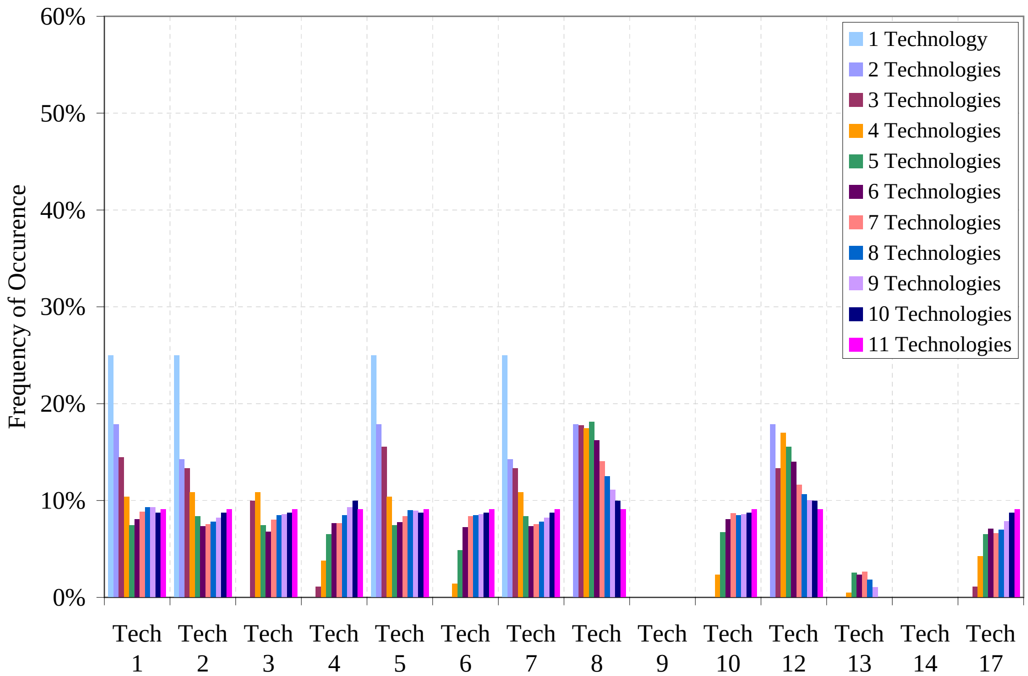


Figure 95: Analysis of the Probability of Occurrence of Candidate Technologies in a Successful HDBT Portfolio.

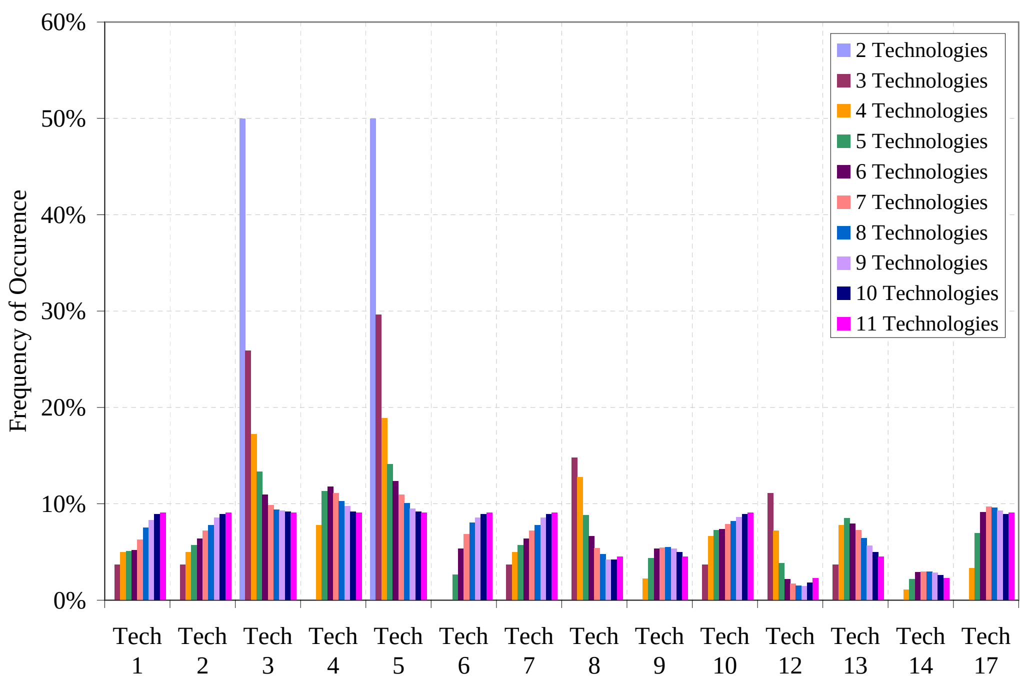


Figure 96: Analysis of the Probability of Occurrence of Candidate Technologies in a Successful Decapitation Strike Portfolio.

5.10.4 Using Inverse Design to Understand the Design Space

Inverse design, a technique for treating any variable as an independent variable described in Section C.4, is a useful means to set targets in top-level capability metrics and identify design solutions that meet multidimensional capability needs. Central to this technique is the use of Monte Carlo simulations to populate the design space with a large number of points and a dynamic graphical visualization of the results. The multivariate analysis feature of the JMP® software package is one of the visualization techniques that will be used in subsequent sections to perform the inverse design technique and identify technology areas using a top-down capability-focused approach.

5.10.4.1 Analysis of Sensitivities for an LRS Campaign Simulation

The first exploratory test of interest is an evaluation of the sensitivities of several MoEs to the available degrees of freedom for a three day LRS campaign. First, a Pareto chart summarizing the impact on the "targets killed" and "platforms lost" metrics is shown in Figure 97.

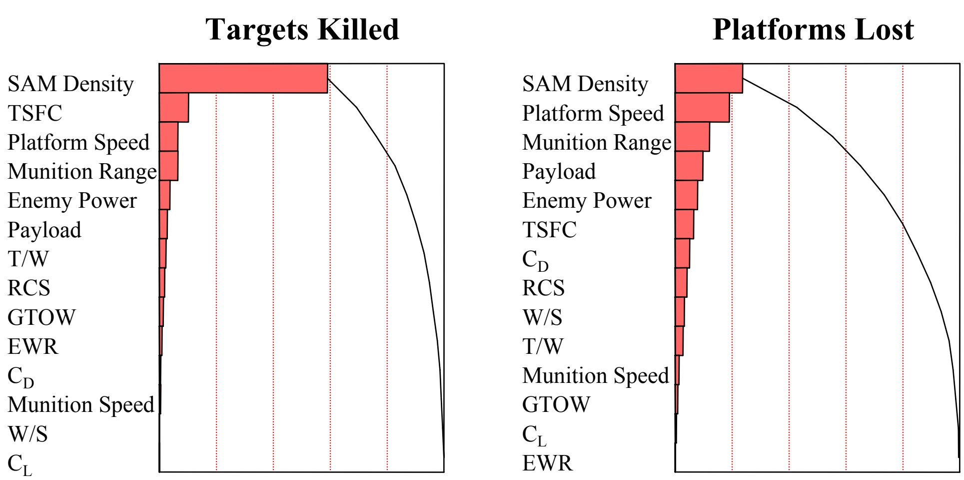


Figure 97: Important Factors on MoEs for the GSTF Three Day Scenario.

The most significant contributor to targets killed is the SAM density. This is because dialing down SAM density controls the guaranteed rate of SAM destruction using Effects-Based Decomposition for SEAD assets. Turning this dial always kills targets. As expected, SAM density is a major contributing factor to platforms lost. A surprising outcome of the data analysis is that within certain bounds, the platform speed is actually a detractor with respect to both MoEs. An analysis of the cases causing this unexpected result revealed that high speed platforms require high thrust leading to high fuel flow rates. This causes the intelligent agents operating the platforms to return to base frequently to refuel, limiting their effectiveness over the 24 hour simulation period. In fact, since their decision logic is not perfect and since unexpected engagements may occur during the egress to base, platforms sometimes run out of fuel while attempting to return to base, contributing negatively toward the "platforms lost" metric. While long range and high speed are desired, this analysis points to the fact that perhaps high speed *dash* as opposed to high speed *cruise* is a desirable trait for an LRS system.

The 4,625 cases executed for the GSTF three day scenario can also be analyzed using the multivariate analysis as shown in Figure 98. A color-coding scheme (ROYGBIV) was used to represent cases where all six platforms were lost as red and no platforms lost as violet. Initially, the entire space was filled with red points that made it difficult to observe the other trends. This is because the three-day scenario is exceptionally difficult and all platforms are lost in 89.3% of executed cases. The data in Figure 98 can be supplemented using a color map on the correlation values between each of the design variables and the MoEs as shown in Figure 99. In this figure, red indicates a perfect positive correlation (as indicated by the red color on the diagonal) and blue indicates a perfect negative correlation. Black boxes are drawn around the variables with the strongest magnitude of correlation. In the GSTF three day scenario case, platform speed, SAM density, and TSFC have the strongest negative impact on targets killed. Interestingly, the color map identifies platforms lost and targets killed as negatively correlated. While these two dependent variables are theoretically independently calculated from each other, they are actually heavily correlated. As more platforms are lost, not only are there fewer LRS assets to prosecute the remaining targets, but the enemy SAM sites have fewer adversaries to attack with a finite number of missiles. This is a quintessential example of the nonlinearity of combat: losing a wingman does not make your job linearly harder, it makes it exponentially harder.

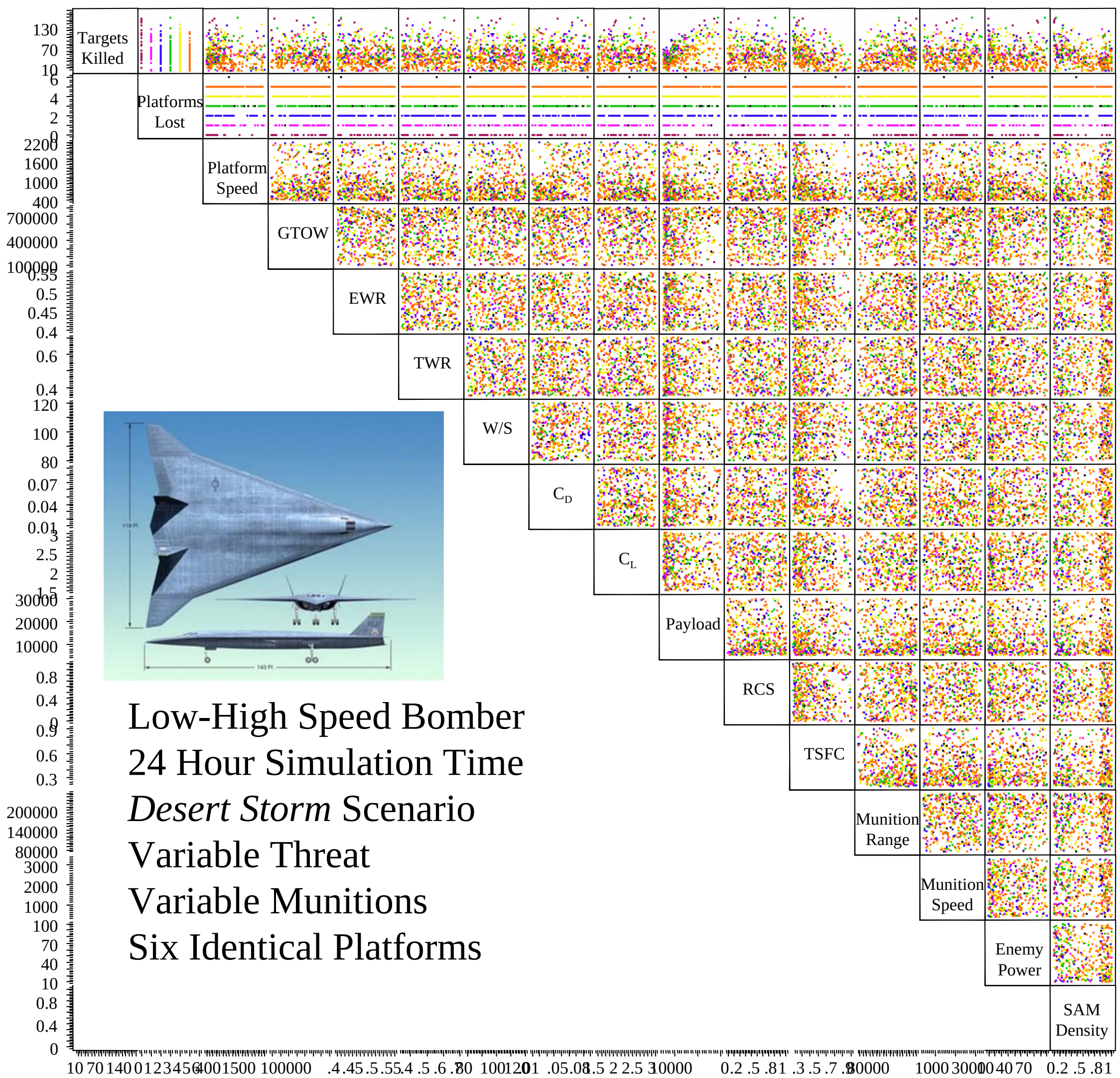


Figure 98: Multivariate Analysis for the GSTF Three Day Scenario.

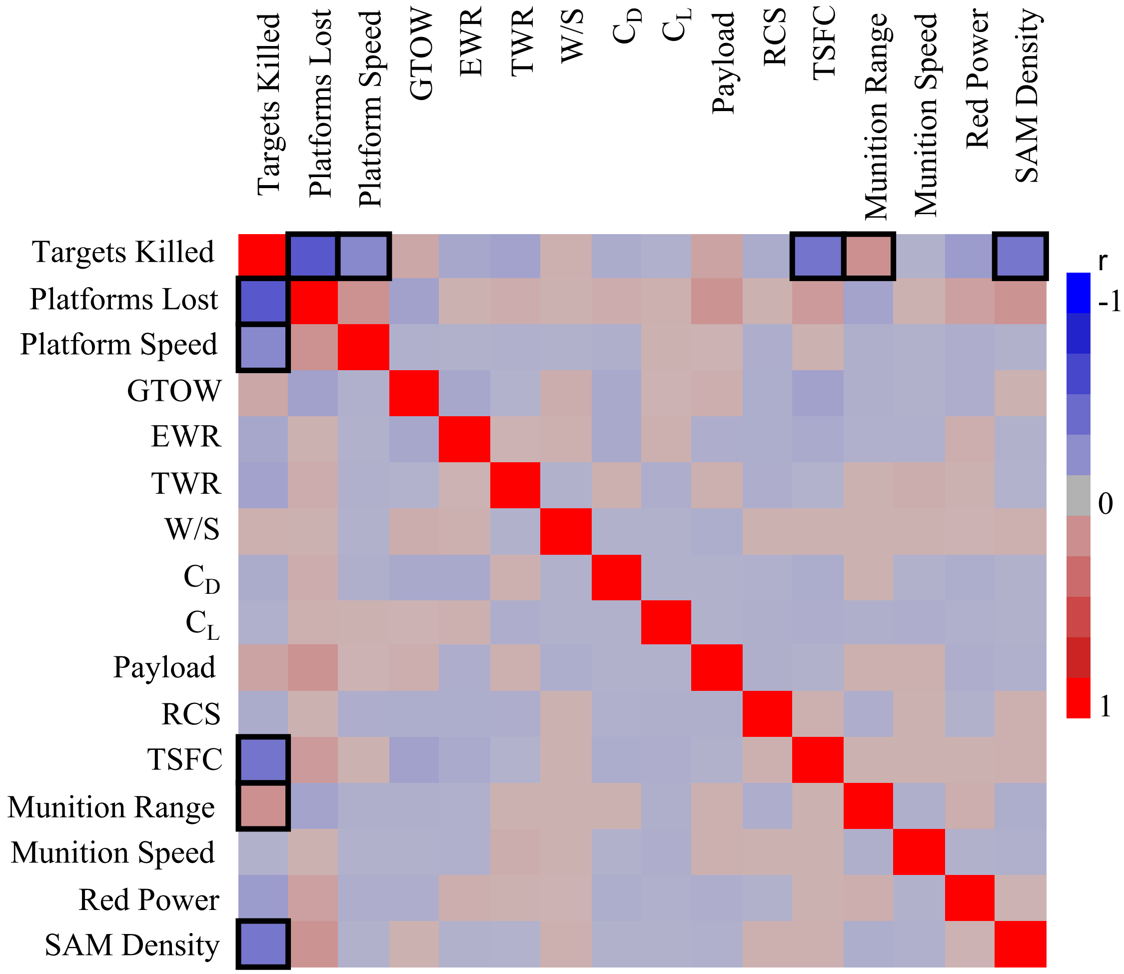


Figure 99: Color Map on the Correlations of Independent Variables and MoEs for the GSTF Three Day Scenario.

In addition to the sensitivity analysis for the individual technology factors, a key attribute in capability-based technology evaluation is the quantification of the ability of proposed technologies to exceed the state of the art. The multivariate analysis showing a comparison between the baseline case and successful cases throughout the design space is shown in Figure 100.

As previously mentioned in Section 5.10.2, the present day baseline configuration described in Table 20 kills 20 targets in eight hours while losing all six platforms. Figure 100 shows 384/4625 cases for the GSTF scenario where more than 20 targets are killed and less than six platforms are lost. The color scheme indicates high losses (hot colors) and low losses (cool colors). The black square in the figure indicates the value of the baseline case. When the black square is surrounded by colored points, this indicates that little technology infusion is needed in that dimension because the 384 cases have similar thresholds of the technology metrics in that dimension. On the other hand, when the black square is far from the centroid of the colored area, an area for technology infusion or design variation from the baseline case is highlighted. For example, the baseline case has high TSFC and low munition range compared to many of the more successful cases.

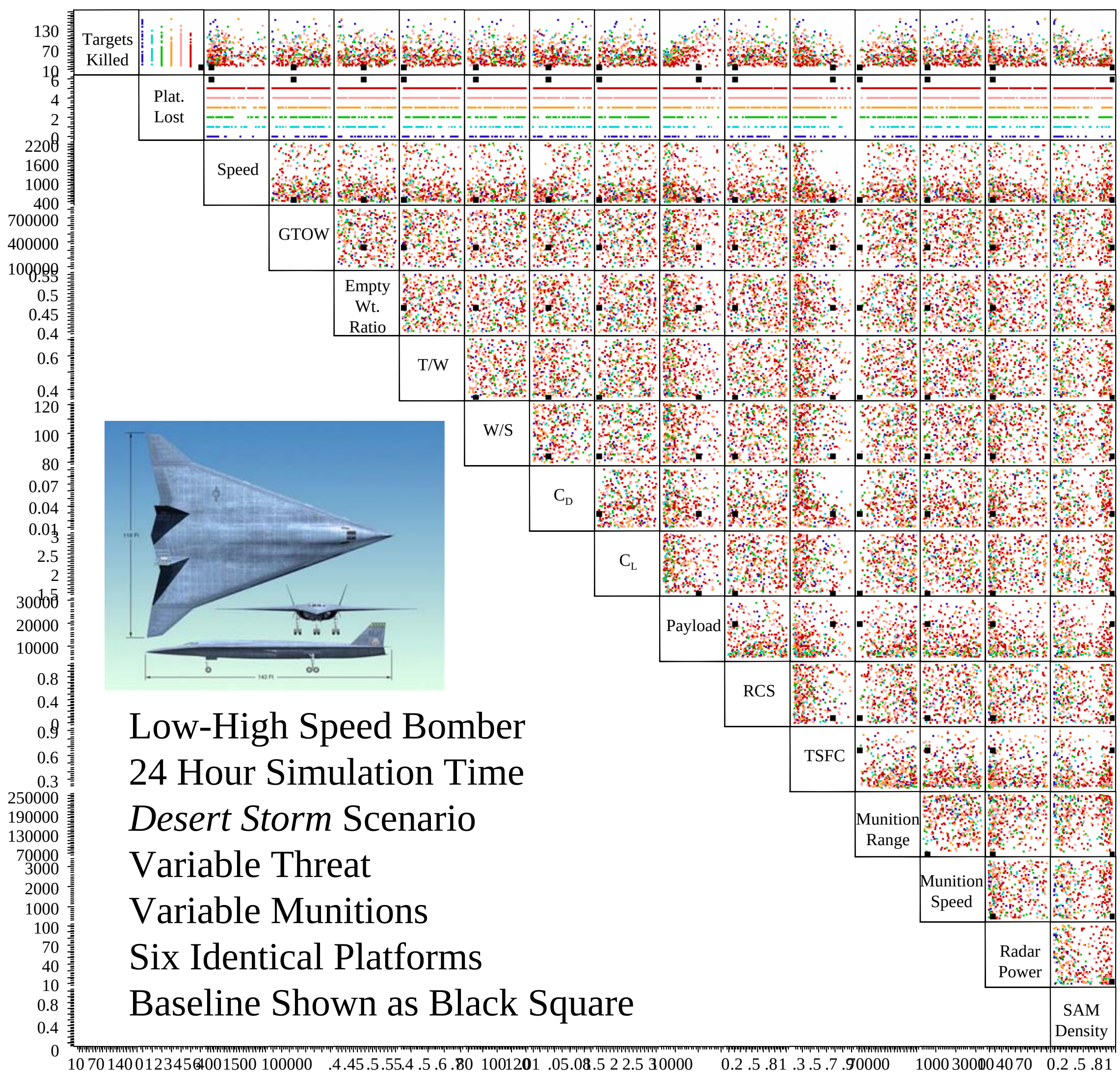


Figure 100: Comparison of the Capability Baseline to 384 Cases that Improve the State-of-the-Art.

The list of comparison points is further narrowed to the 22 cases for which more than 20 targets are killed with zero platform losses in Figure 101. The same observations as in the previous figure are true for Figure 101 as well with a slight caveat: most of the successful solutions occur when the SAM density is very low or near-zero.

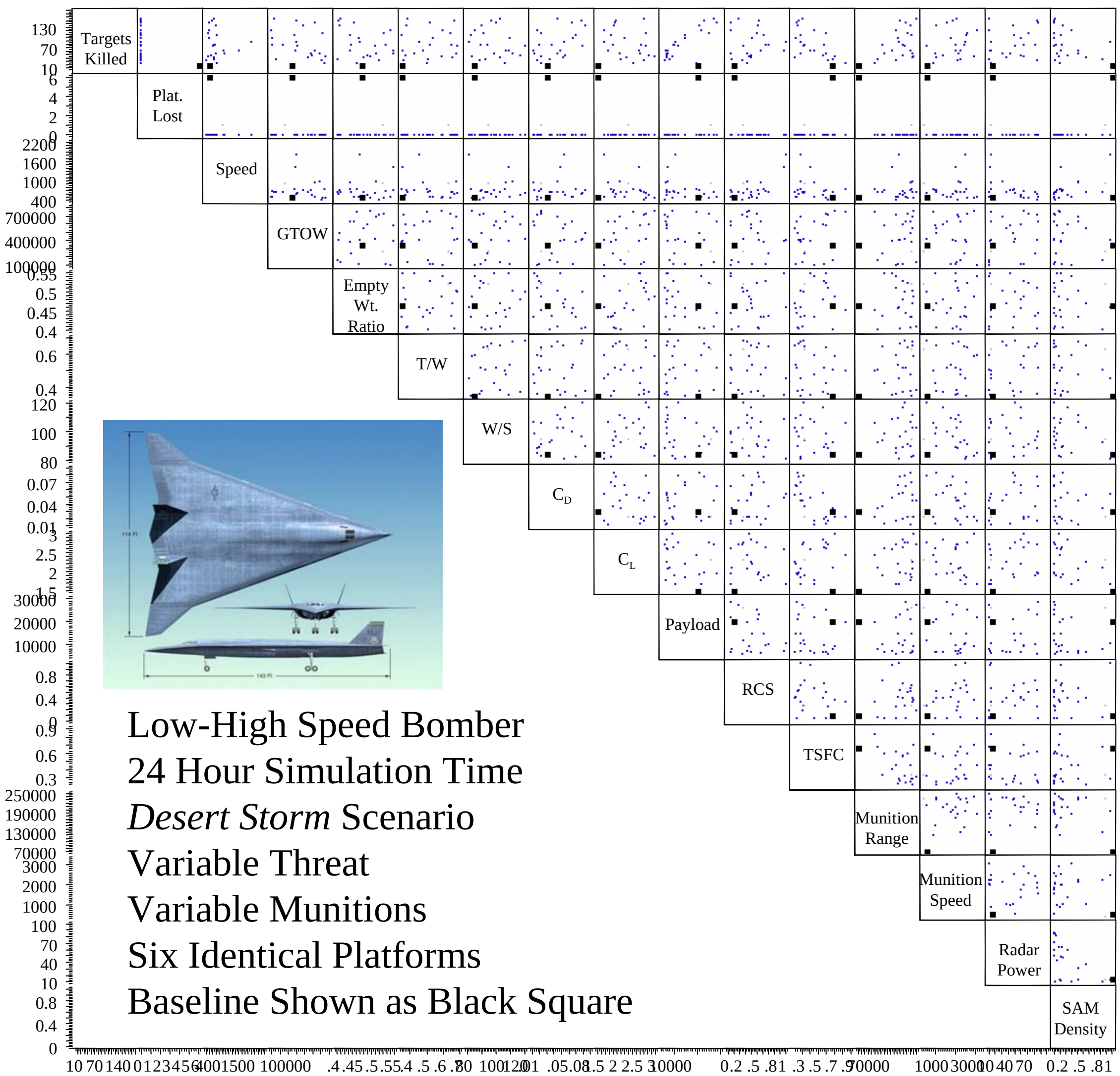


Figure 101: Comparison of the Capability Baseline to 22 Best-in-Class Cases.

The distribution plots in Figures 102 and 103 enable better visualization of these trends. The first figure shows how the technology factors and design variables are distributed for the 384 "good" designs that improve on the capability baseline. These designs tend to have smaller payloads, speeds less than Mach 2, long range munitions, and low fuel consumption.

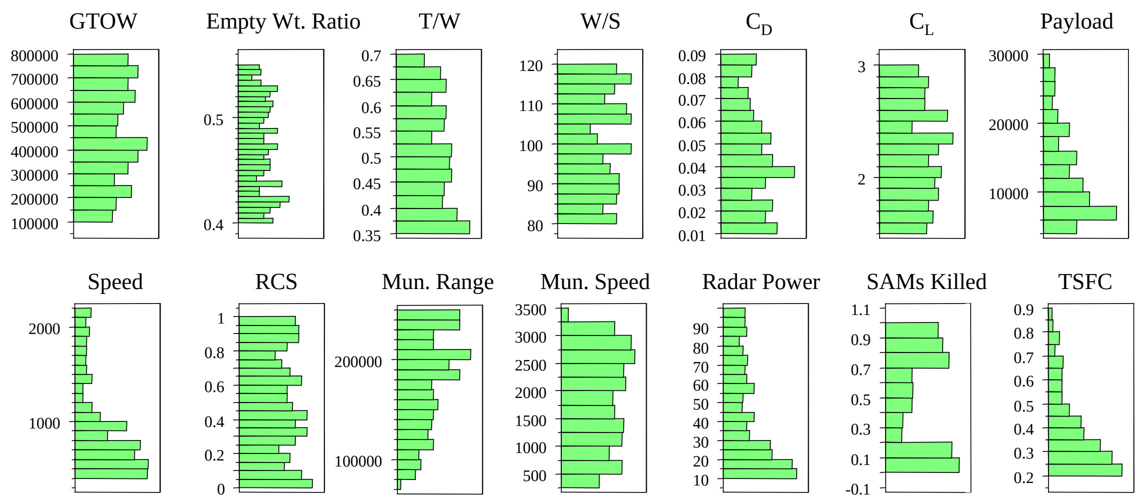


Figure 102: Distributions of Design/Technology Factors 384 Good GSTF Designs.

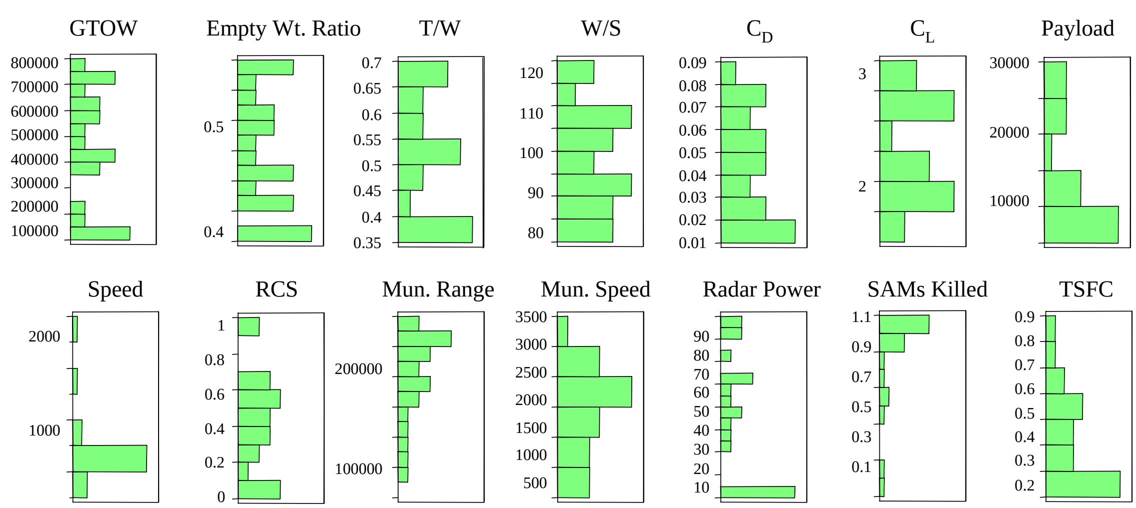


Figure 103: Distributions of Design/Technology Factors for 22 of the Best GSTF Designs.

Figure 103 shows how these trends evolve when only the 22 best-in-class solutions are considered. Notably, the speed for these points generally falls in line with the "sweet spot" of about 750 knots that is explained in the next section. These aircraft also tend to use long range munitions and have good fuel consumption; however, they are all also operating in a benign threat environment. This underscores both the value and difficulty of parametric analysis: while the variability of the threat adds flexibility, narrowing to a specific answer is difficult.

Fortunately, the neural networks generated for the GSTF scenario can be used to generate large volumes of additional data while holding some factors constant. A Monte Carlo simulation of 1,000,000 points was executed using uniform distributions on all design parameters and arbitrarily setting the SAM density to 50% and the Enemy Radar Power to 15 dB. Of the 1,000,000 cases, those with losses of 100% and kills below a threshold of 20 (defined by the performance of the baseline case) were eliminated to leave 88,258 Monte Carlo runs. Several interesting trends emerge when these cases are analyzed. First, the dataset can be divided into those solutions in which 0, 1, 2, 3, 4, and 5 platforms are lost respectively.

The zero loss and five loss case are further examined in Figures 104 and 105 respectively. While the GTOW, Empty Weight Ratio, and Thrust-to-Weight ratio of the zero loss case is biased toward solutions with higher values, the results for the five loss case are fairly uniformly distributed. Lift coefficient has little variation for either case. Curiously, it appears that solutions with higher drag coefficients are preferred in the cases when zero platforms are lost. The opposite is true for less successful engagements. Both solutions tend to favor long munition ranges, but there is a definite preference to a certain value of payload weight in the zero loss case. The same preferred range for speed emerges again in this analysis. There is also a clear preference for solutions with very low radar cross section and thrust specific fuel consumption in the zero loss case.

Of the 88,258 GSTF Monte Carlo runs with a fixed threat and platform losses less than six, 3,987 cases can be identified for which the number of platforms lost is zero. These cases are depicted using the multivariate profiler in Figure 106

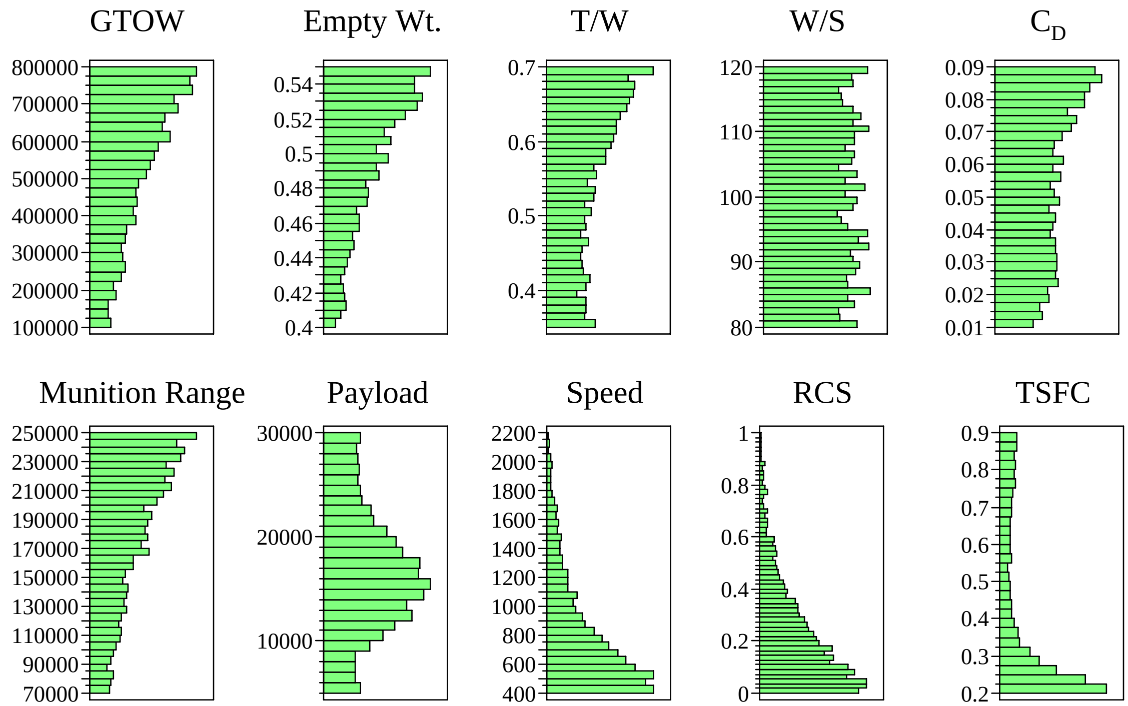


Figure 104: Distribution of Design Variables for a Fixed Threat, Zero Loss Case for the GSTF Scenario.

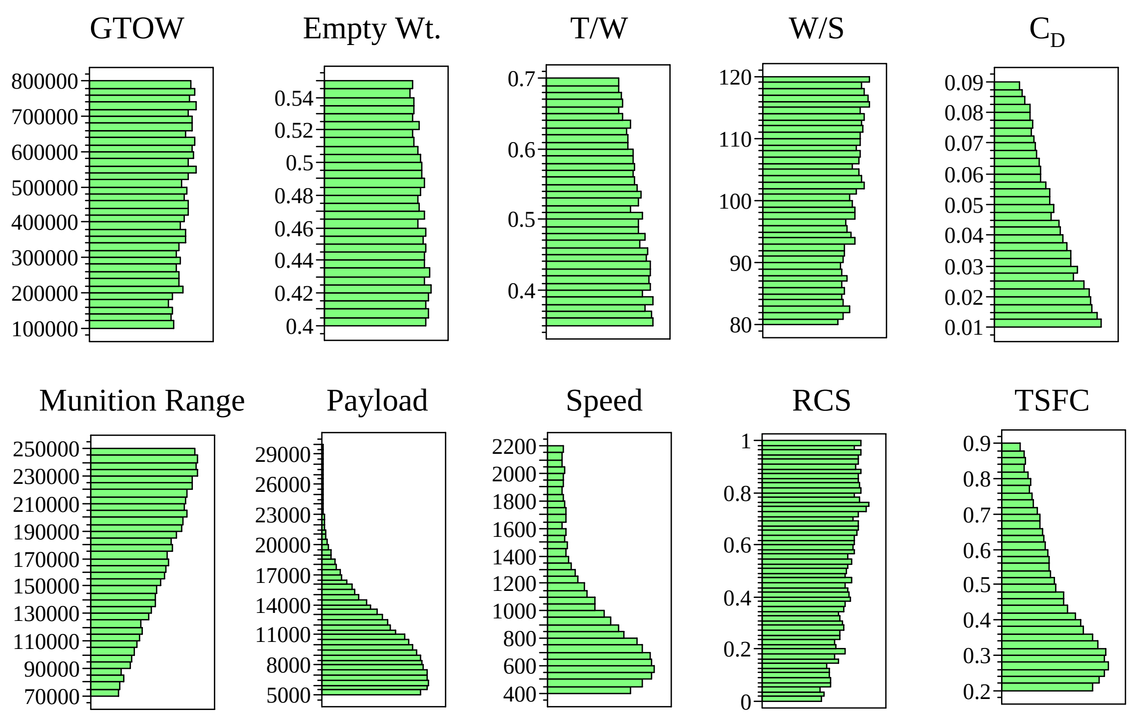


Figure 105: Distribution of Design Variables for a Fixed Threat, Five Loss Case for the GSTF Scenario.

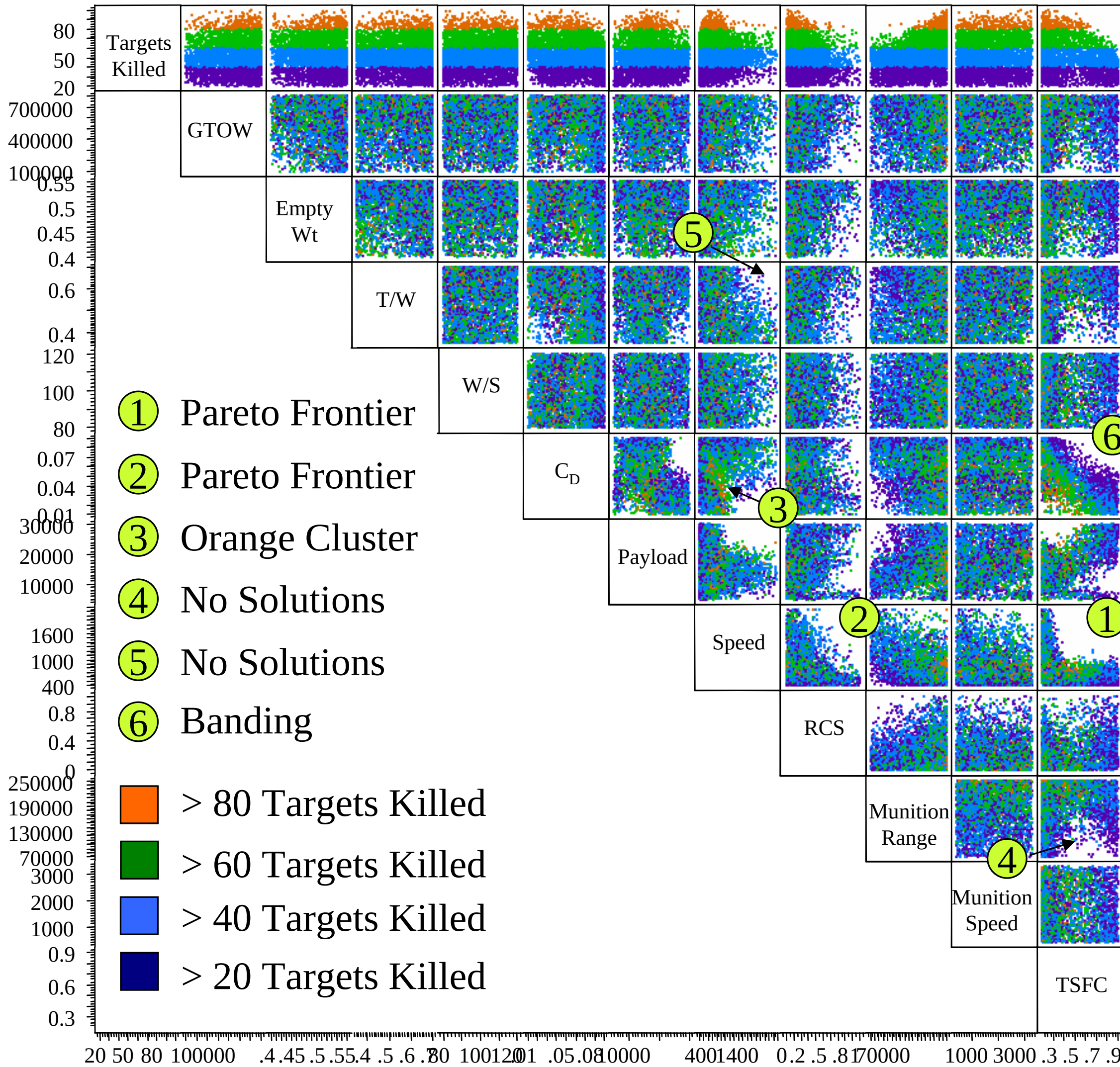


Figure 106: Multivariate Profiler for 3,987 Zero-Loss, Fixed Threat, GSTF Cases.

In this figure, the color banding represents the number of targets killed and is grouped in units of 20 as shown in the figure legend. Several interesting behaviors are seen in this plot. First, an interesting Pareto frontier is visible in area 1. Here, solutions with either low TSFC, low speed, or some combination of the two are valid. A similar Pareto frontier is visible in area 2 between speed and RCS. The lack of points in the upper right corner of this plot indicates that aircraft with high cruise speed and high RCS are unsuccessful. Since high speed actually has a positive impact on survivability, the observed trend must a secondary effect related to the low GTOW tendency of these solutions. In area 3, highly desirable solutions in terms of targets killed are located in the center of the region with less effective solutions radiating outward from the range of speed values observed throughout this work. There is an area of maximum effectiveness that matches a particular drag coefficient to a certain speed regime. Curiously, there are no solutions in regions 5 and 6. The void in region 5 is attributed to high T/W and high speed cases. Failures in this region are a function of the high fuel burn of these solutions. While the plot shown in region 6 has a higher concentration of cases in the desirable upper left corner, it is interesting to note that some marginally effective solutions are viable for aircraft with poor fuel consumption at all munition ranges. Finally, in area 6, an interesting banding pattern is observed. This plot is magnified in Figure 107 where it is evident that the most effective solutions are those with low drag coefficient and low fuel consumption. The different bands identify thresholds of effectiveness and illustrate a primary dependence of effectiveness on the correct definition of these two parameters. The red "X" in Figure 107 indicates the area of least effectiveness while the yellow star highlights the desirable region.

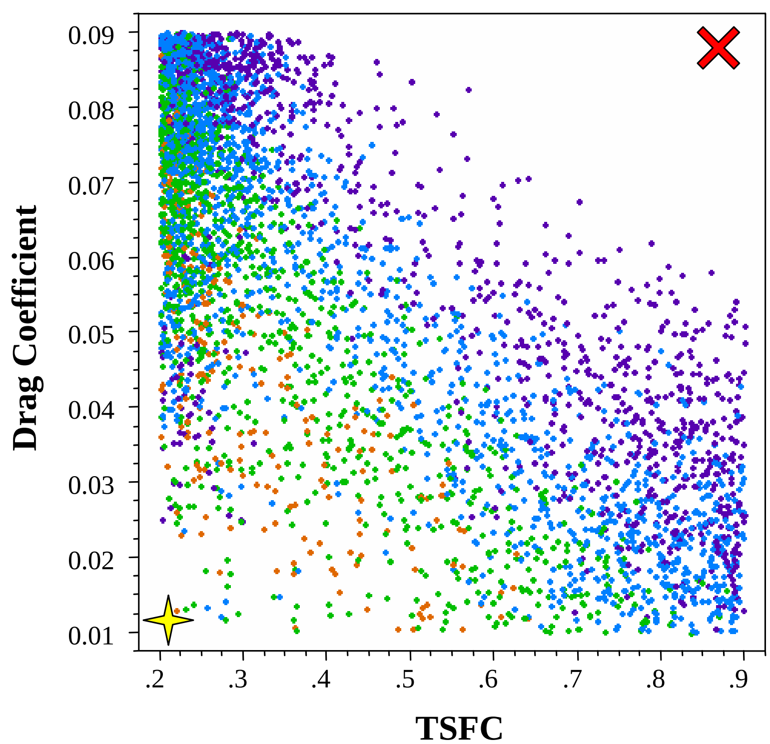


Figure 107: Relationship Between Drag Coefficient and TSFC for Zero-Loss, Fixed Threat, GSTF Cases.

This section highlights sensitivity analysis for a three-day simulation of many interoperating assets. While the technology-related parameters that contribute to effective capability delivery cannot be precisely identified with ease, the parametric tradeoff environment provides a graphical means for identifying the sensitivities of various technology and design variables on each other. Furthermore, the analysis of data does not stop with the execution of cases and the generation of surrogate models. Due to the many degrees of freedom and the layers of complexity, graphical visualization techniques must be used to "tell the story" that supports design decisions. The power of surrogate models for rapid manipulation of data and the use of probabilistic techniques to generate off-design cases were demonstrated for a single scenario with some assumptions fixed. It is easy to see how the scope of the problem quickly grows out of control.

To further illustrate how to use the SOCRATES method for decision making, subsequent sections analyze specific "analysis questions" that arise in the development of a Long Range Strike system architecture. The surrogate model-enabled parametric tradeoff environment is queried in different ways and various graphical tools are used throughout to develop answers to these questions.

5.10.4.2 *Identifying the Thresholds for Speed and Fuel Consumption*

In a 2005 interview, General Jumper said "higher, faster, farther... is not an absolute measure. We have to figure out where it's useful to us... Can it carry a practical payload? Can it get there and return? Can you take advantage of all this velocity" [411].

High speed efficient propulsion systems are a focus area for AFRL and other technology development organizations. The SOCRATES method can be used to identify requirements for speed and TSFC using modeling and simulation. Using the surrogate models created for the GSTF three day attack scenario, a 5,000 case Monte Carlo simulation was executed holding all factors constant at their baseline values except the platform speed and platform TSFC. The resulting multivariate analysis, supplemented with an inset three-dimensional profiler is shown in Figure 108.

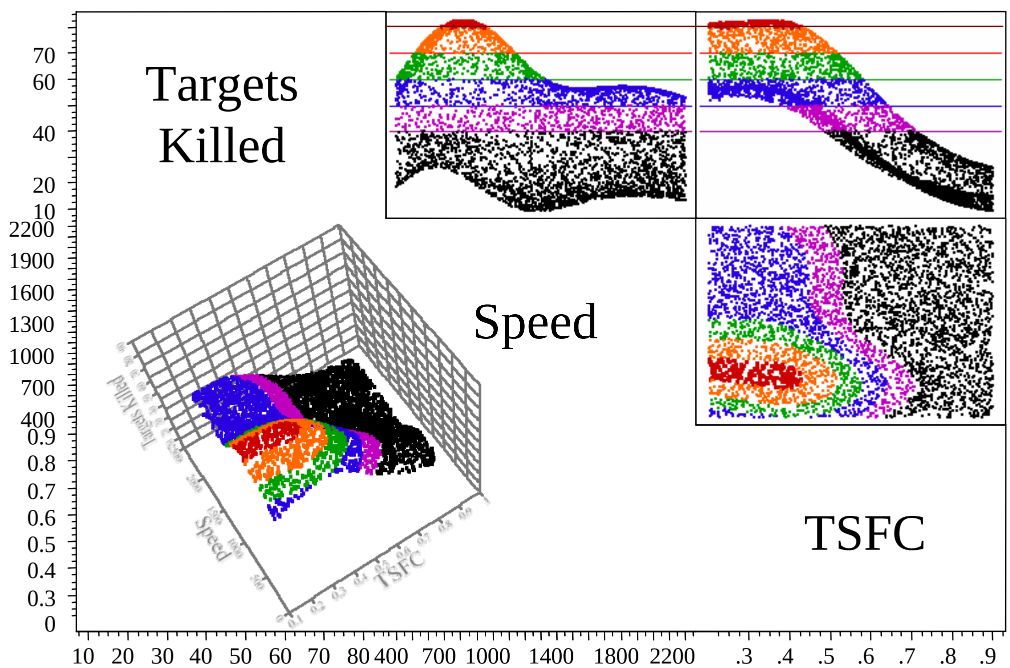


Figure 108: Multivariate Analysis for Speed and TSFC for the GSTF Three Day Attack Scenario.

In this figure, cases where more than 80 targets are killed are color-coded red. The other colored areas represent 70 targets killed (orange), 60 targets killed (green), 50 targets killed (blue), and 40 targets killed (purple). As shown in the multivariate analysis, there is a "sweet spot" for speed and TSFC for this mission in this scenario with all other factors held constant. This region of interest migrates around the design space as other factors are changed. A three-dimensional contour profiler illustrating dynamic visualization is shown in Figure 109. The Z-axis of the contour plot is the predicted number of targets killed and the X and Y-axes are speed and TSFC respectively. A nominal case where the inputs of the surrogate models are set halfway between their minimum and maximum ranges is shown in Figure 109. Using the parametric slide bars to the right of the figure, the character of the design space can be quickly queried for variations.

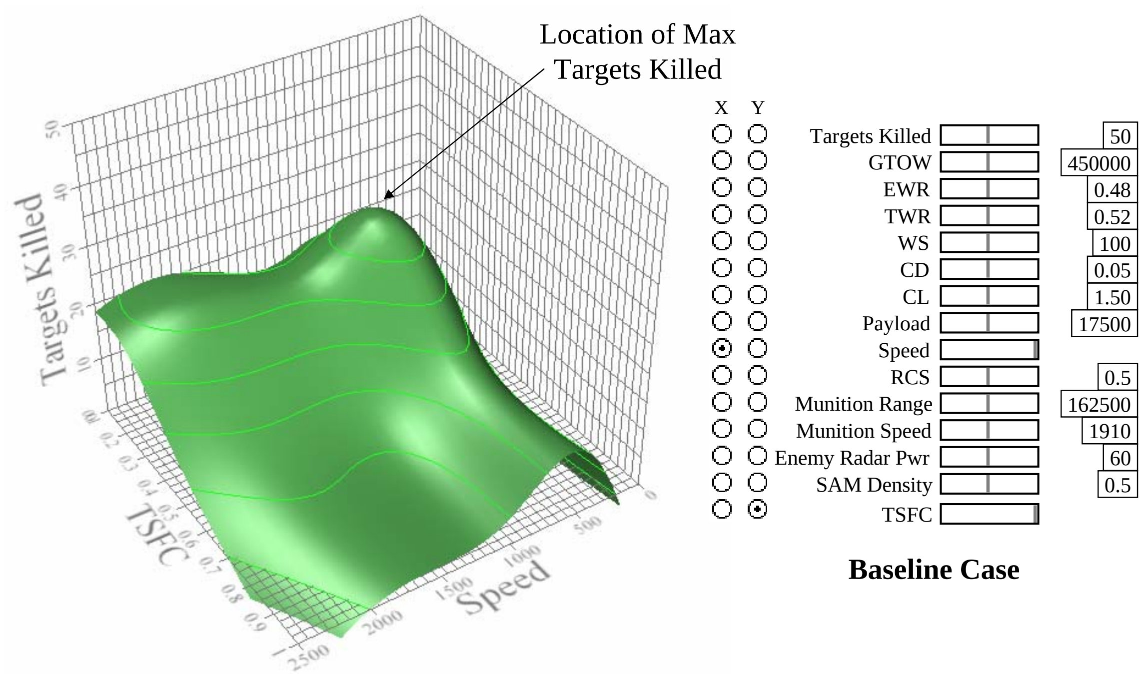


Figure 109: Interactive Three Dimensional Contour Plot Showing a Nominal Case.

In general, solutions with high TSFC values and speeds beyond a certain value are not attractive in terms of targets killed. This is a function of the range of GTOW values used, limited fuel volumes for the payloads employed, the unrefueled range required in the simulation, and the 24 hour time limit imposed on the simulation.

The subsequent figures illustrate how the inputs to the surrogate model can be interactively changed to produce new three dimensional contours on the fly[29]. An increase in thrust-to-weight ratio is shown in Figure 110. Here, the maximum point for targets killed

[29] As always, these results are prefaced by the disclaimer "For the scenario used and under the assumptions considered."

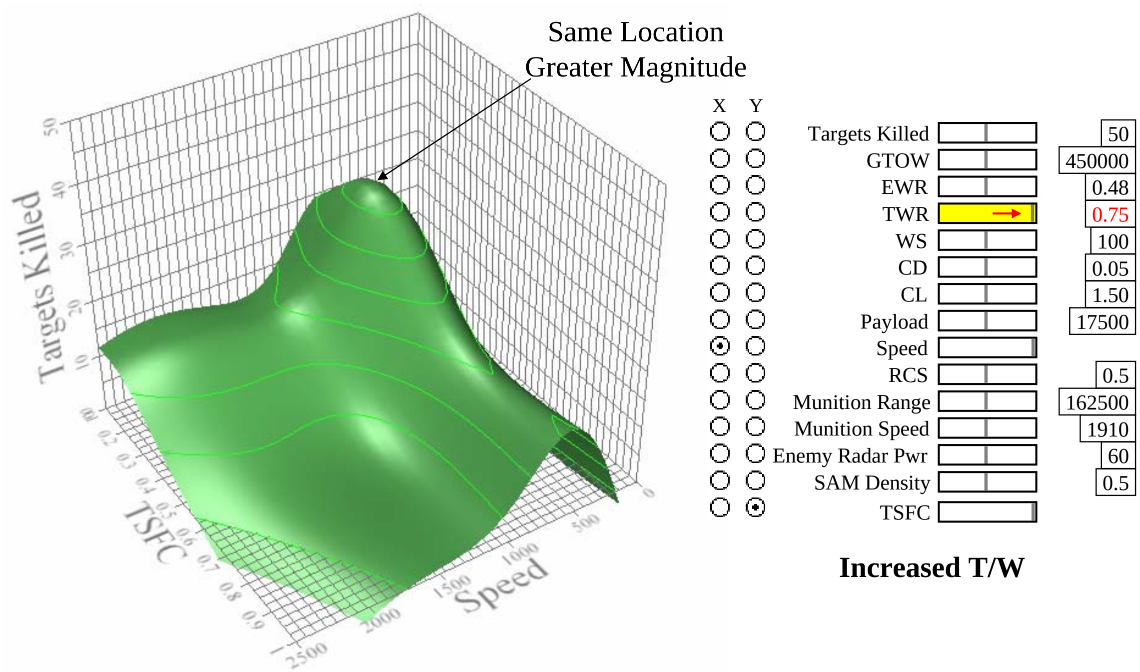


Figure 110: Interactive Three Dimensional Contour Plot Showing an Increase in Thrust/Weight Ratio.

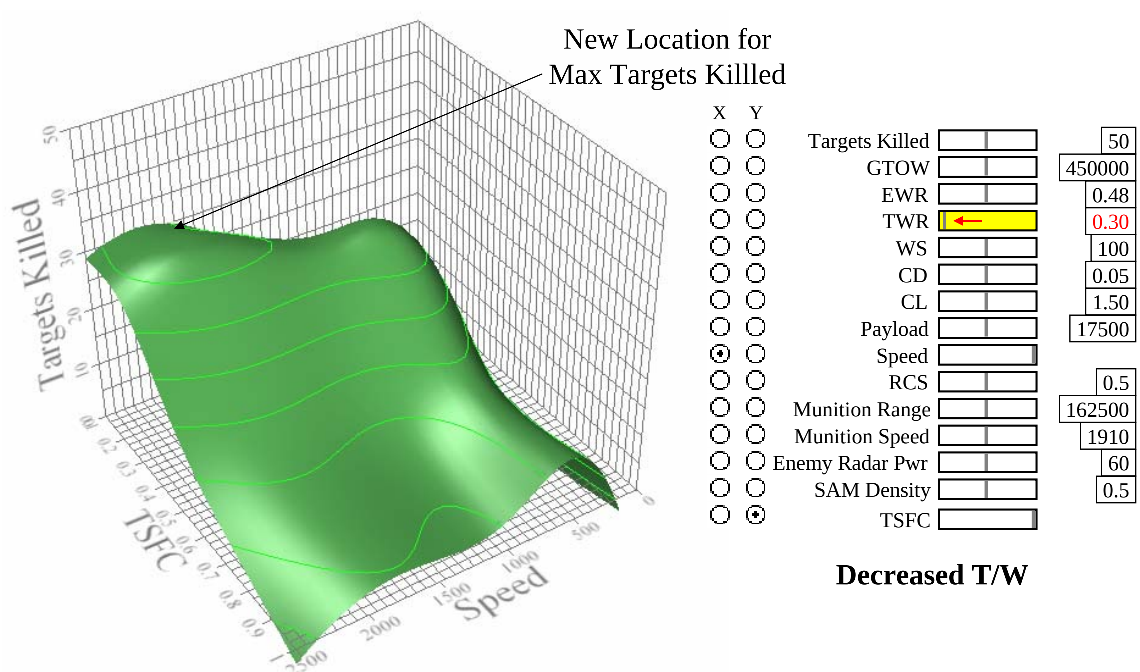


Figure 111: Interactive Three Dimensional Contour Plot Showing a Decrease in Thrust/Weight Ratio.

and platforms lost remains at the same X-Y coordinates but increases in magnitude. In contrast, as illustrated in Figure 111, a decrease in thrust-to-weight ratio decreases the first peak and shifts the maximum point to a different locus in the speed vs. TSFC space. This new point is in the supersonic regime with extremely low TSFC. Essentially, for a fixed GTOW, vehicles with a lower thrust-to-weight ratio have less thrust available and hence burn less fuel at maximum power regardless of TSFC. If such low TSFC values could be obtained through technology infusion or a change in the propulsion system architecture, supersonic solutions become attractive.

Figure 112 shows how the character of the space changes as short range munitions are used. As the munition range is decreased to around 15 km (8 nm or 50,000 ft), the magnitude of targets killed decreases across the design space. While the munition range has no direct impact on the probability of kill, this tradeoff demonstrates the coupling between MoEs. When munition range is lower, platforms must ingress closer to the target before weapon release. This exposes them to more SAM sites, increasing the number of platforms lost. As more platforms are lost, the number of targets killed decreases in a nonlinear manner. There is also a secondary effect: long range weapons are released sooner. The sooner a weapon is released, the sooner that platform can be retasked to a new engagement, which contributes to increasing target kills.

To further illustrate the power of the technique, in Figure 113 the SAM density is decreased from 50% to 15%. With fewer SAM sites to destroy platforms, the number of targets killed increases to its previous level. Although the bombers operate with short range munitions, the lower SAM density decreases their likelihood of being killed and therefore increases the overall effectiveness of the solution.

The final permutation illustrated in Figure 114 further examines the parametric threat by changing the power of the enemy's radar from 50 dB to 15 dB. As with the previous example, a decrease in radar power has a similar effect to lowering the SAM density. The magnitude of the "targets killed" response increases dramatically.

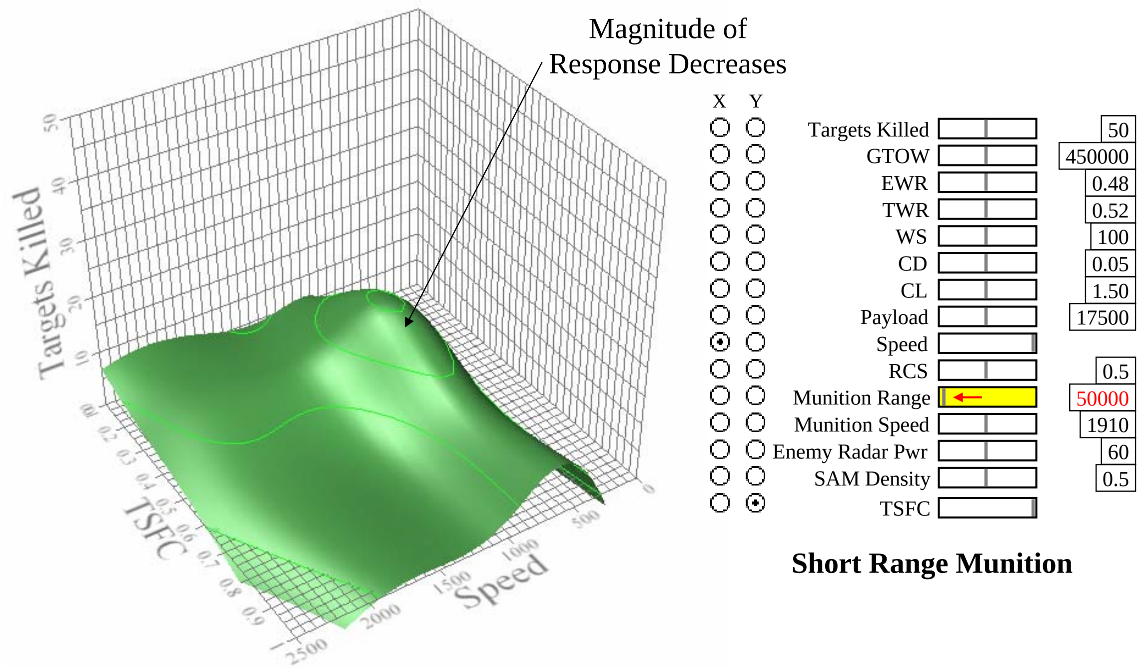


Figure 112: Interactive Three Dimensional Contour Plot Showing a Decrease in Munition Range.

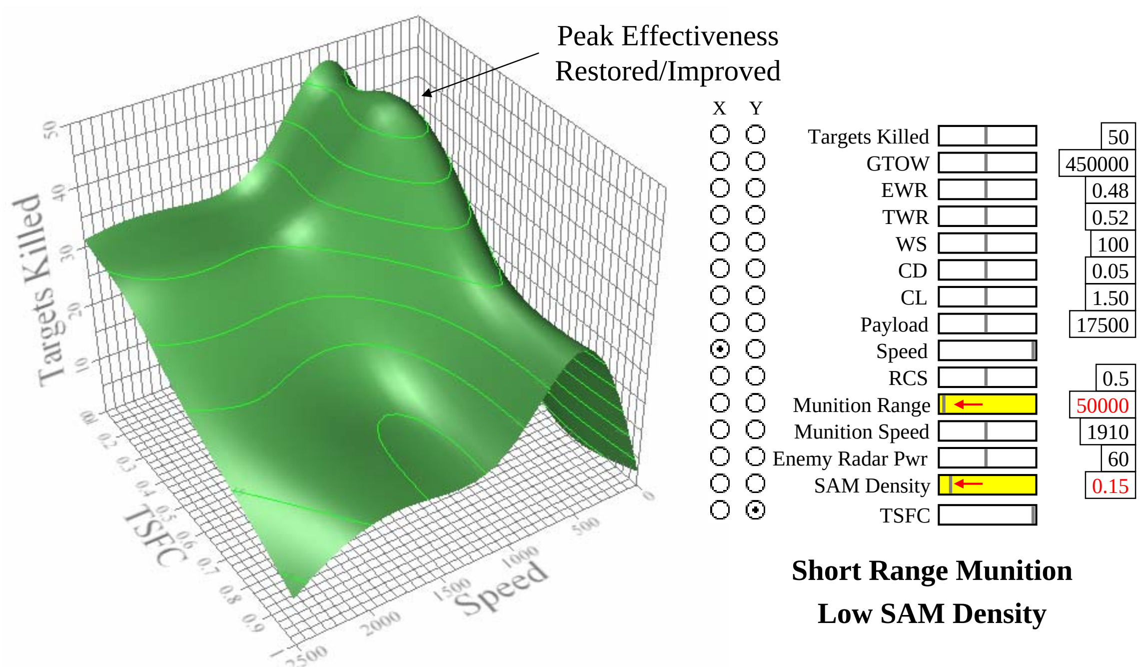


Figure 113: Interactive Three Dimensional Contour Plot Showing a Decrease in Munition Range and SAM Density.

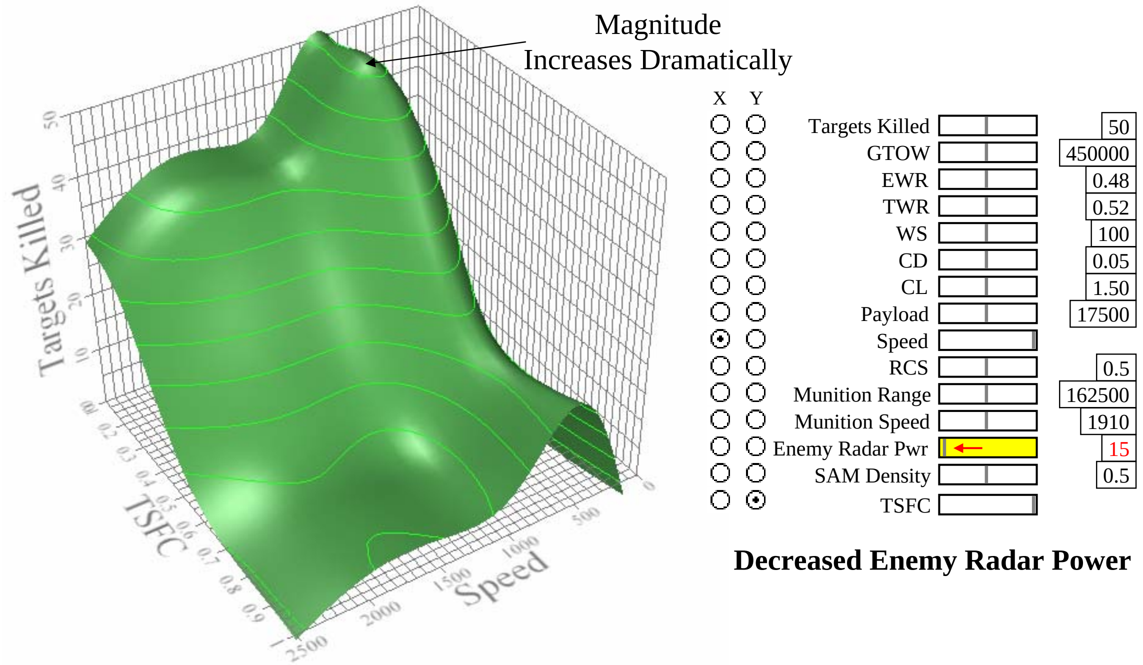


Figure 114: Interactive Three Dimensional Contour Plot Showing a Decrease in Enemy Radar Power.

5.10.4.3 How Large and How Fast?

Two key requirements for a future LRS system are the size and speed of the vehicle. Concepts from a 40,000 lb lightweight fighter to a behemoth air vehicle with a GTOW approaching runway limits of 1,200,000 lbs have been proposed. Similar variation from subsonic to hypersonic bombers have been advocated for the LRS mission. The surrogate models can be queried to address these tradeoffs as a function of other design variables, assumptions, and threats. A tradeoff between gross takeoff weight and platform speed using the JMP® contour profiler is shown in Figure 115. In this example, the contour profiler is used to evaluate the predictive neural networks for targets killed and platforms lost for the HDBT and decapitation strike scenarios. Constraints are applied to color-code regions where either a platform is lost or no targets are killed for both scenarios. The white space therefore indicates a feasible design region where both missions can be accomplished without any platform losses. As a caveat, the "answer" yielded by this study is a function of the assumptions shown in Table 30.

The contour shown in Figure 115 shows that the design space is constrained on the upper portion by excessive gross weight which makes it impossible to fly the distances required for the C_L and C_D values given. The left side is constrained by the speed for the decapitation strike mission: below approximately Mach 1.6, this aircraft cannot kill any targets regardless of the GTOW. At the lower ranges of GTOW, the available fuel is so low after the empty weight ratio and payload weight are taken into account that the platform cannot complete the mission without running out of fuel. This is highlighted in the contour profiler as all four constraints are active at low GTOW.

The contour profiler also allows the values of any variable to be changed in real-time, which means that the resulting contours can take any number of shapes depending on how the assumptions of the problem are varied. To illustrate the dynamics of the problem, all design variables and assumptions are held constant except for the value of the drag coefficient. While the previous paragraph stated that the aircraft cannot fly at very high gross weight values with the drag coefficient given, this statement was actually a theory that was tested by varying the contour profiler to determine what behavior was causing the infeasible region to appear. When the drag coefficient is lowered dramatically by 50%, this constrained area disappears as shown in Figure 116. This trade study illustrates the value of decreasing platform drag. When the drag is lowered, the constraint on the HDBT Platforms Lost recedes. Since the empty weight ratio and TSFC are unaltered, the amount of fuel and the fuel flow rate are essentially held constant. When the drag is reduced, the thrust required is reduced. Since the overall fuel burn decreases as thrust decreases (also observed as an increase in L/D in the Breguet Range Equation), the platform range increases and the likelihood of losing the platform due to fuel depletion decreases. As a result, for a constant cruise speed, a platform of lower GTOW is viable.

This analysis was also performed for the GSTF Attack Three Day Scenario and summarized in Figure 117 and the design space differs greatly. This example shows the maximum feasible area for the GSTF Attack case. Here, there is a "sweet spot" of GTOW and speed. Since the platform flies at its maximum speed throughout the mission, excessive platform speeds have a negative impact on mission success due to increased fuel burn.

Table 30: Assumptions and Design Variable Ranges for the Analysis in Figures 115 and 116.

Variable	Figure 115	Figure 116
Empty Weight Ratio	0.482	0.482
Thrust/Weight Ratio	0.60	0.60
Payload (lbs)	30,500 lbs	30,500 lbs
W/S	81 lb/ft^2	81 lb/ft^2
C$_D$	**0.078**	**0.039**
C$_L$	2.06	2.06
P$Kill$	70%	70%
Radar Cross Section	0.38 m^2	0.38 m^2
Weapon Speed	Mach 0.85	Mach 0.85
Weapon Range	295 km (160 nm)	295 km (160 nm)
Fan Pressure Ratio	2.2	2.2
Overall Pressure Ratio	26	26
Turbine Temp (T4)	1697°K (3054°R)	1697°K (3054°R)
Enemy Radar Power	22	22
Enemy SAM Density	86%	86%

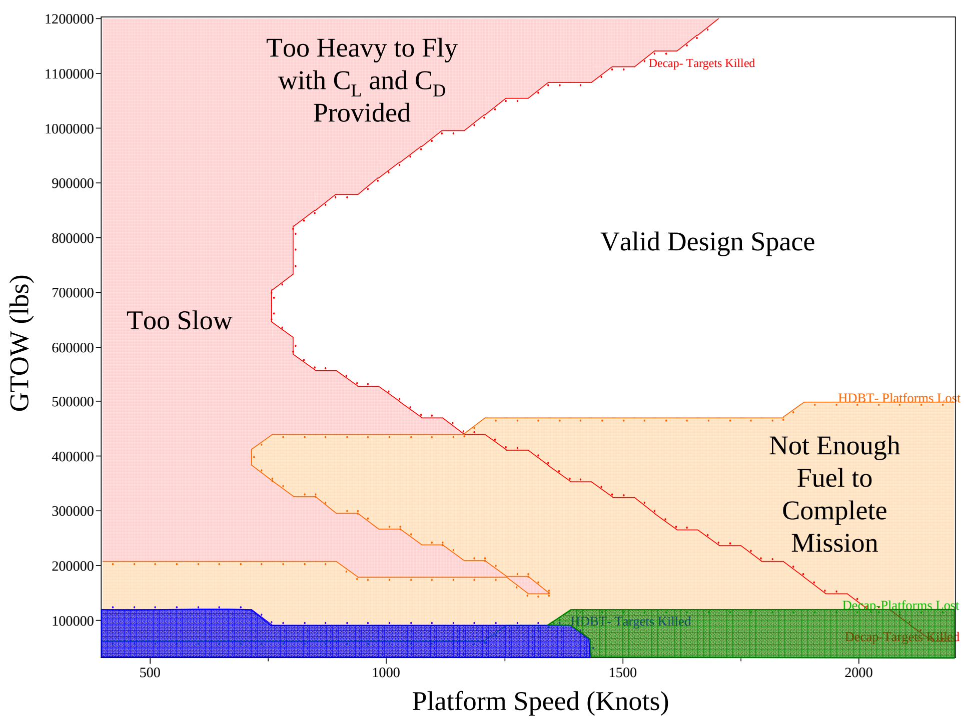


Figure 115: Contour Plot of GTOW vs. Speed for the HDBT/Decapitation Strike Mission.

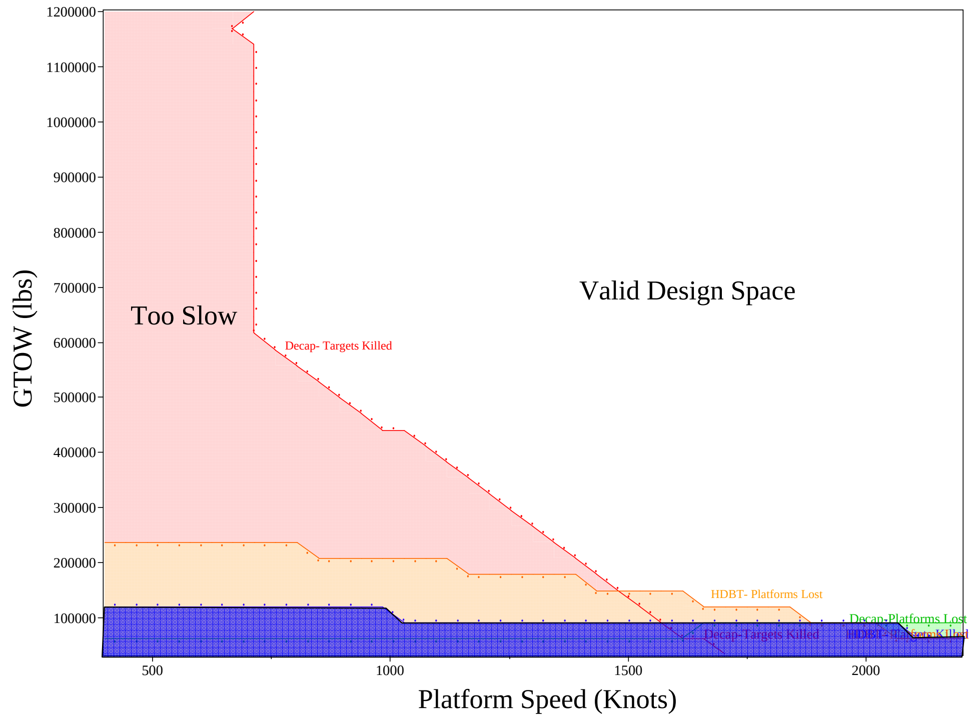


Figure 116: Contour Plot of GTOW vs. Speed for the HDBT/Decapitation Strike Mission (50% Reduction in Drag).

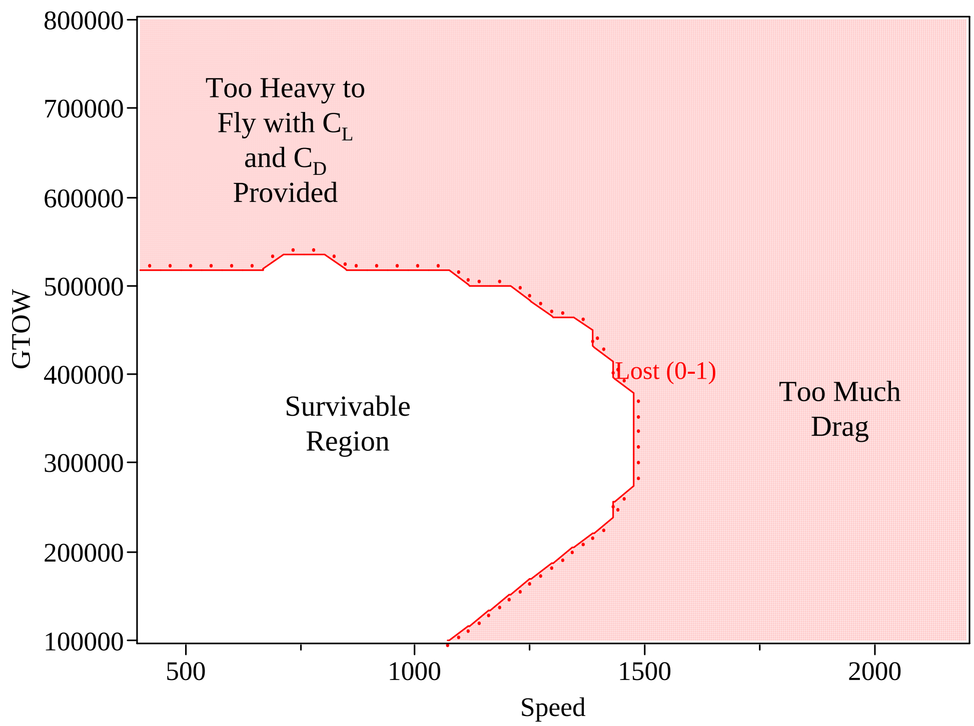


Figure 117: Contour Plot of GTOW vs. Speed for the GSTF Attack Mission.

5.10.4.4 Allocation of Speed and Range: Platform or Munition

Another trade involves the allocation of speed to either the platform or the munition. Where does speed have the greatest benefit on the number of targets killed? Figure 118 shows the distribution of platform speed for the HDBT and decapitation strike cases when the target is killed and no platforms are lost. The distribution of successful engagements for the HDBT case is entirely uniform, indicating that there is no preference toward platform speed when the simulation time is five hours. On the other hand, when the simulation is constrained to ninety minutes in the decapitation strike case, the distribution of points shows a bias toward higher speeds: the lower part of the distribution is missing. Further analysis indicates that the points from this part of the distribution occur for unsuccessful engagements.

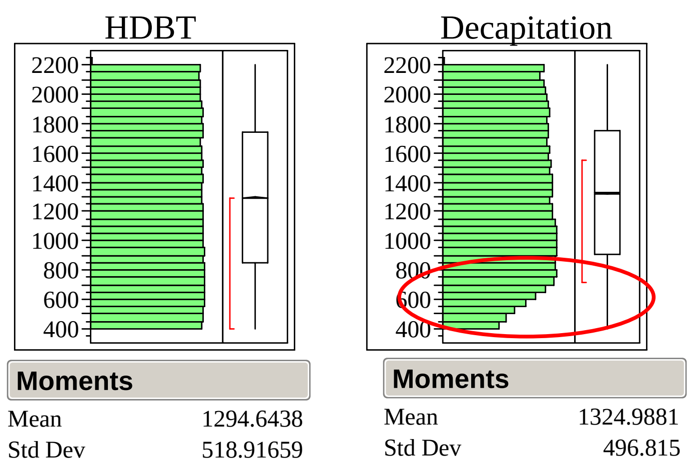


Figure 118: Impact of Platform Speed on the HDBT and Decapitation Strike Scenarios.

Additionally, it is desired to examine the relative importance of platform speed versus weapon speed. How should technology development resources be allocated between a high-speed platform and a high-speed, long-range munition? This tradeoff is illustrated by taking the same data and examining the distribution of points with respect to munition speed and range for the HDBT and decapitation strike cases as shown in Figure 119. Again, the distribution is cut-off, indicating that there tends to be a bias toward higher speed munitions for the decapitation strike case. While this analysis identifies trends, it is not possible to use these plots to identify either where the speed should be allocated or under what conditions a high speed weapon provides value.

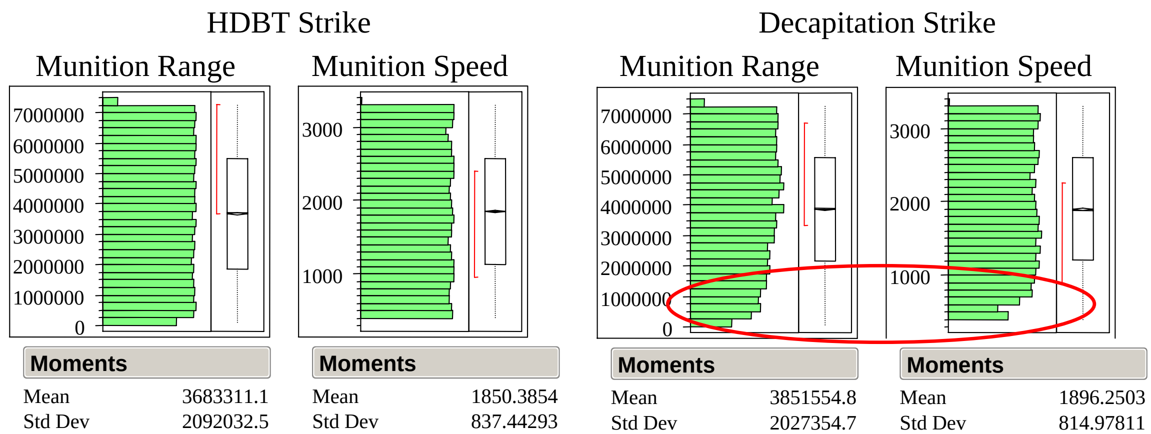


Figure 119: Impact of Munition Range and Speed on the HDBT and Decapitation Strike Scenarios.

To further analyze the situation, a composite capability metric called "Success" was devised to identify cases where targets killed was equal to 1 and platforms lost was equal to zero. Using the ANOVA procedure, a Pareto chart that indicates the dominant factors contributing to a successful engagement can be identified for the HDBT and decapitation strike scenarios as shown in Figure 120. As illustrated in the figure, the most important factor in a successful engagement is a long munition range. This is due to the fact that a standoff weapon keeps the platform out of the SAM envelope. If the munition range is sufficiently long, the design of the platform has no impact on survivability because the platform never comes under attack. When the surrogate models are queried, for very long standoff ranges, the platforms lost metric is always equal to zero. This tends to highlight the value of the missileer solution introduced in Section 2.4.3.

The Pareto chart on the left of Figure 120 shows that platform speed is the fourth most important variable, accounting for about eight percent of the variability of the response. The GTOW and payload weight plays a significant role in the HDBT mission due to the ability of the platform to carry the heavy loads anticipated. Munition speed has almost no impact on the success of the HDBT mission. This is not true for the decapitation strike mission, as the munition speed is the third most important parameter for a successful engagement. As the Pareto chart indicates, the importance of munition speed is actually greater than that of platform speed, but not by an appreciable amount.

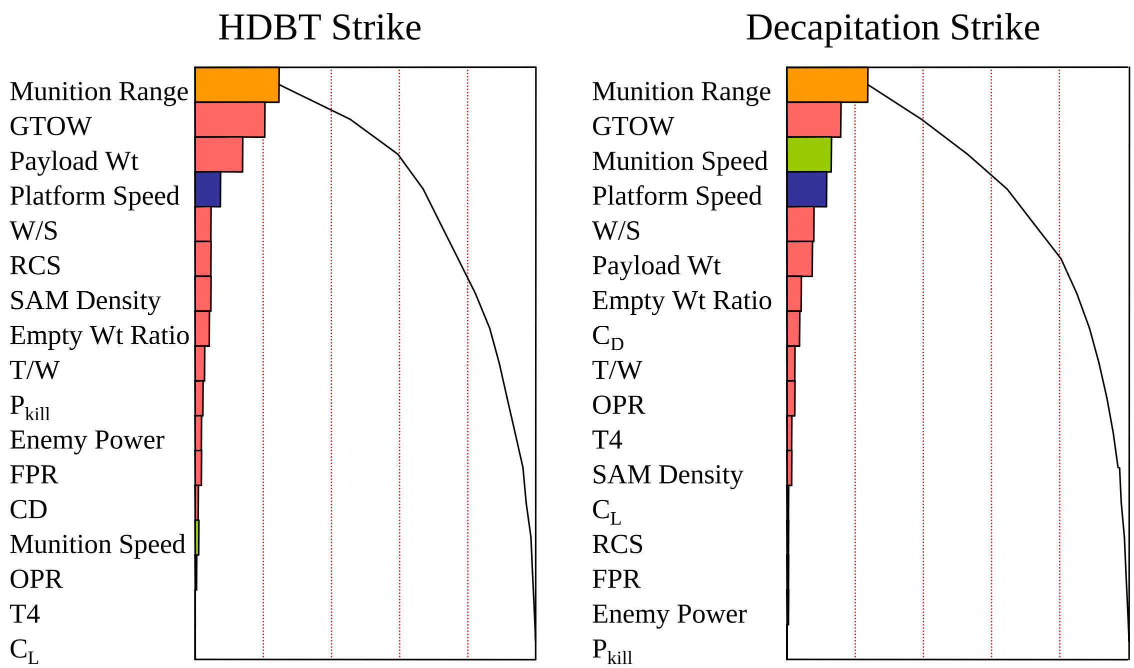


Figure 120: Analysis of Variance for Allocation of Speed Between the Platform and Munition.

This answer changes significantly as the variability due to munition range is removed from the picture. In Figure 121, the weapon is fixed as a short range bomb. In this case, the most important factor for the HDBT mission is the density of enemy SAM sites, which obviously impacts the "platforms lost" element of successful engagements. In the decapitation strike scenario, when the weapon is fixed as a short range bomb, the platform speed becomes the dominant factor. Figure 122 shows the results when the weapon range is fixed at 1,500 km (810 nm). For the HDBT case, munition speed now becomes a dominant factor only because the two most important platform variables remain GTOW and payload weight and all other factors relate to the range and survivability of the platform. Since the weapon is released far outside the range of the threat, the impact of the remaining factors is minimal. In the case of the decapitation strike, both the platform speed and munition speed are near the top of the list in the Pareto chart. While it is interesting that the platform speed and munition speed change roles between the two plots, further analysis indicates that the relative importance of these two factors is highly dependent on the settings of munition range and fluctuates wildly as the platform range is varied.

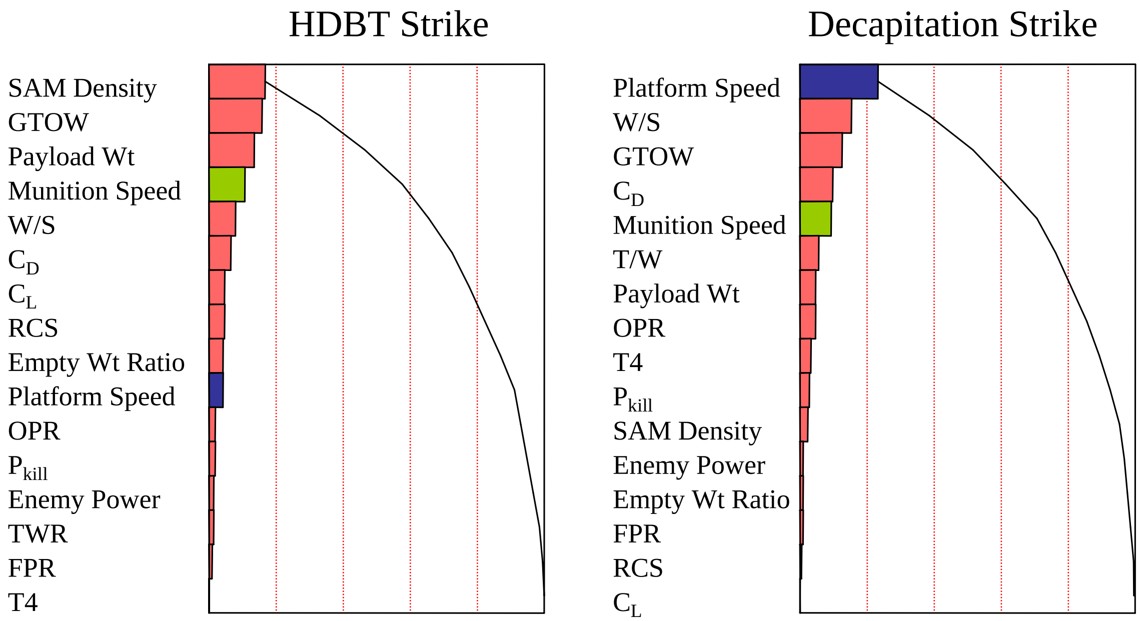


Figure 121: Analysis of Variance for Allocation of Speed Between the Platform and Munition (Fixed Weapon, Short Range).

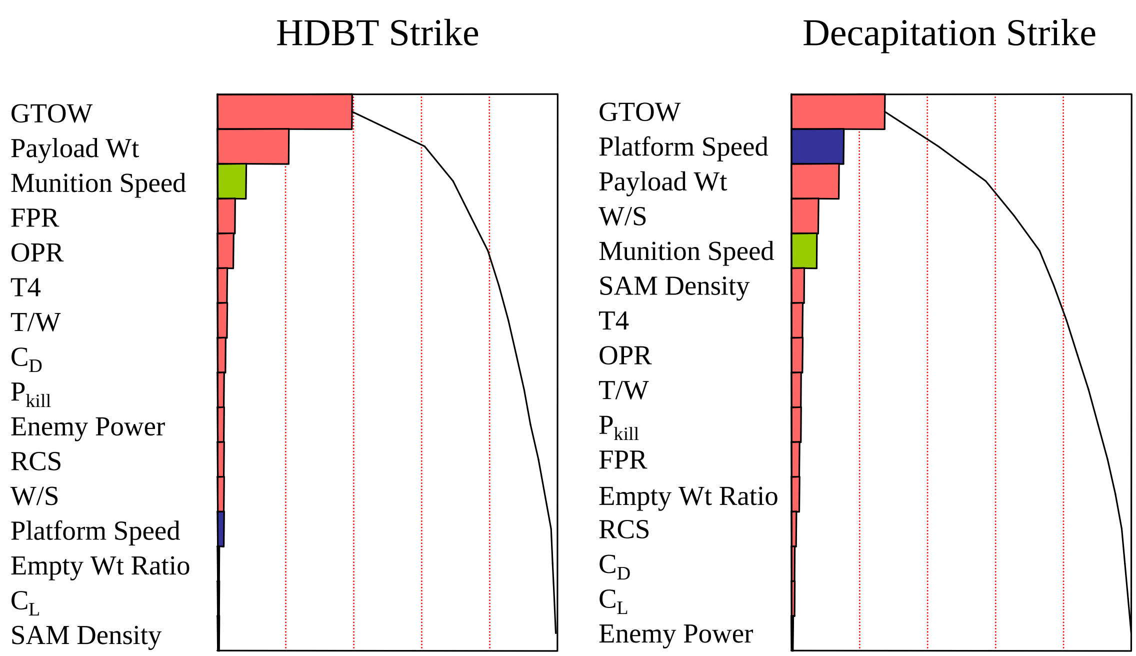


Figure 122: Analysis of Variance for Allocation of Speed Between the Platform and Munition (Fixed Weapon, Long Range).

This trade study illustrates one of the fundamental results of this work. When technology metrics are analyzed with respect to top-level capabilities using modeling and simulation, definitive ***answers*** are few and far between. The are so many degrees of freedom in the problem that certain dimensions must be temporarily locked to recognize any discernable pattern. This phenomenon is akin to the concept of the total derivative. While a partial derivative is a well defined trend that can be viewed with respect to all other variables held constant, when all variables are allowed to change over large ranges, the result is a trade space where essentially any answer can be obtained provided that certain design variables and assumptions are set to desirable levels. To this end, the aforementioned graphs provide some information about the character of the trade space, and while the dynamic tradeoff environment enabled by surrogate models is useful for demonstrating the type of information that can be gleaned from these models, presenting "snapshots" of the design space without a detailed understanding of the underlying assumptions can be extremely misleading to the decision maker.

The allocation of speed and range between platform and munition was also examined for the GSTF three day scenario as shown with a contour profiler in Figures 123 and 124. Note that the axes on Figure 124 have been reversed to make the figure more readable. In these figures, solutions that use long range high speed munitions are in the upper left corner. In Figure 123, this region corresponds to the maximum in terms of targets killed. This region is also where the fewest platforms are lost as shown in Figure 124. The stepwise character results from the use of neural network equations to match this discrete response. While these figures illustrate the benefit of long range high speed munitions, there is no clear "knee in the curve" where the behavior jumps from one region of effectiveness to another. The contour plots shown in these two figures provide a detailed measure of the sensitivity of the response, but do little to quantify the exact values of the requirements needed to meet capability thresholds due to the number of assumptions that go into their creation.

The Pareto chart in Figure 97 indicated that platform speed has a greater impact than munition speed in the scenario examined. Using the interactive dynamic environment, the Y-axis can be easily changed to depict platform speed instead of munition speed. The

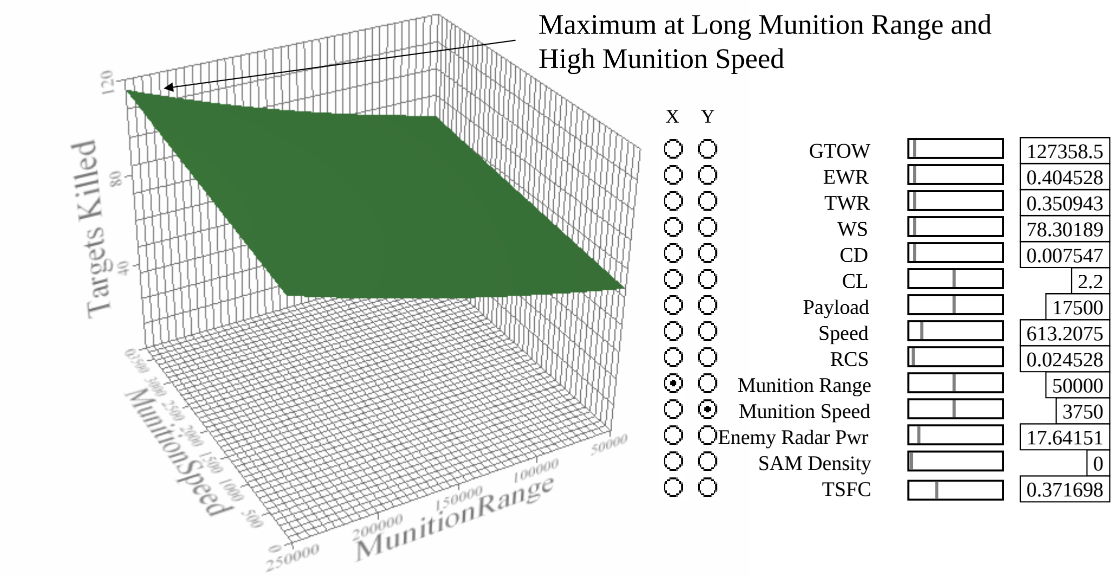


Figure 123: Contour Plot of Munition Speed vs. Munition Range for the GSTF Three Day Scenario (Targets Killed).

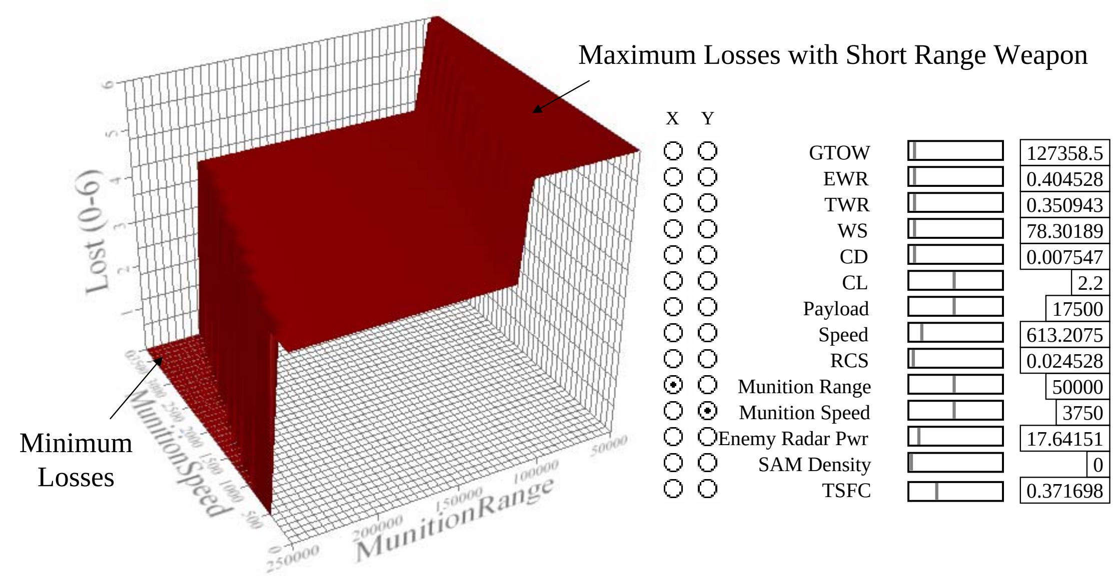


Figure 124: Contour Plot of Munition Speed vs. Munition Range for the GSTF Three Day Scenario (Platforms Lost).

contour plots for the targets killed and platforms lost responses are shown in Figures 125 and 126 respectively. In contrast to the relatively flat surface observed in the relation between munition speed and range, when the design space is examined in terms of platform speed an interesting trend emerges. There is a region around approximately 750 knots where platforms are most effective at prosecuting targets. This "sweet spot" is based on the conditions of the scenario and the characteristics of the threat; however, it is also related to the physics of the problem. The thrust required can be calculated for a given speed and drag coefficient and related to the design variables by:

$$Thrust = Drag = \frac{1}{2}\rho V^2 \left(GTOW\right)\left(\frac{W}{S}\right)^{-1} C_D \tag{19}$$

Fuel burn is a function of TSFC and thrust at a given flight condition. The behavior in Figure 125 therefore reflects the fact that higher speed platforms require more thrust, burn more fuel, and therefore have a limited range for a fixed fuel volume. The squared term in Equation 19 defines the behavior in this figure.

Figure 126 exhibits the stepwise character observed previously due to the discrete nature of this response. The "sweet spot" for platform speed is also reflected as a region of low platform losses for long range munitions. The increase in platform losses in the left of the figure is due to the same behavior reported above for increasing speed. Additionally, in the upper right corner of the figure the response increases again due to the combination of low speed platforms with short range munitions that cannot survive against the modeled threat.

The highlighted value of long range munitions is not a surprising result; however, the identification of a specific range of platform speed that results from the interplay of fuel burn and the threat is an interesting conclusion. This provides an excellent example of how non-intuitive behaviors can be identified and explained using the SOCRATES methodology.

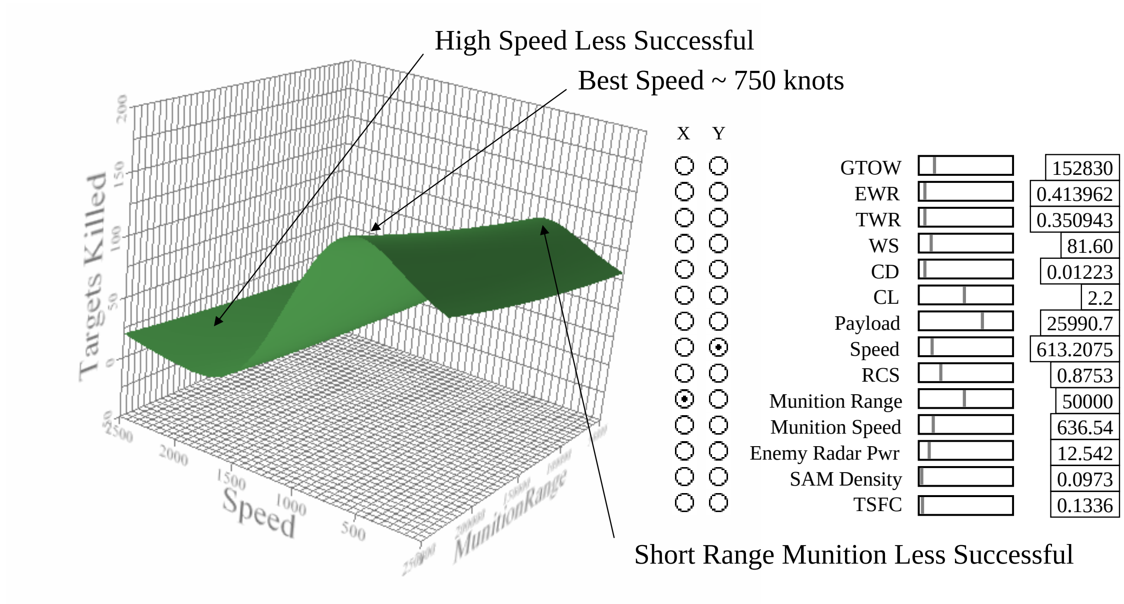


Figure 125: Contour Plot of Munition Speed vs. Munition Range for the GSTF Three Day Scenario (Targets Killed).

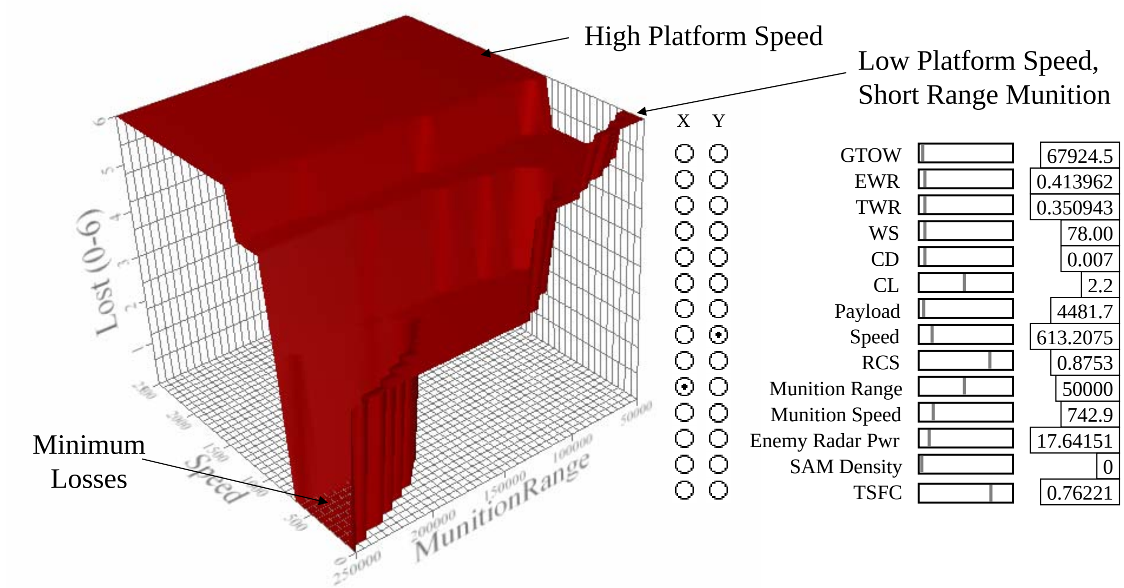


Figure 126: Contour Plot of Munition Speed vs. Munition Range for the GSTF Three Day Scenario (Platforms Lost).

5.10.4.5 *How much survivability is enough? An analysis of Speed and Stealth.*

A fundamental trade that has plagued aircraft designers since the advent of stealth technology in the 1970's is the allocation of speed and stealth to the platform. This trade is confounded by a myriad of other factors, including the design of the weapon, engine, and the parameters of the threat in question. The assumptions used to demonstrate a tradeoff between speed and stealth are given in Table 31.

Table 31: Assumptions and Design Variable Ranges for the Speed vs. Stealth Comparison.

Variable	**Value**
Empty Weight Ratio	0.55
Thrust/Weight Ratio	0.45
W/S	85 lb/ft^2
C_D	0.05
C_L	2.25
P_{Kill}	70%
Fan Pressure Ratio	2.25
Overall Pressure Ratio	27.5
Turbine Temp (T4)	1722°K (3100°R)
Enemy Radar Power	55
Enemy SAM Density	100%

With these degrees of freedom fixed, the total derivative of the trade study can still take on any value as shown in Figure 127. In this figure, successful engagements are colored blue and unsuccessful engagements are colored red. The only discernable trend from this figure is that for very low GTOW values and high payload values, the aircraft design is not feasible (no room for fuel) and hence the engagement is unsuccessful. When the munition and platform variables change simultaneously, no trends are clearly visible. To overcome this difficulty, the tradeoff between speed and stealth was performed for three fixed weapon cases for the decapitation strike mission. In the first case, a subsonic short range bomb is used. In the second case, a long range (610 km/324 nm) subsonic cruise missile is used. The final case holds the range of the cruise missile at 610 km and increases the speed of the missile to Mach 3.75. These three cases are color-coded as blue, green, and red in Figure 128.

Only successful engagements for the decapitation strike mission are depicted in Figure

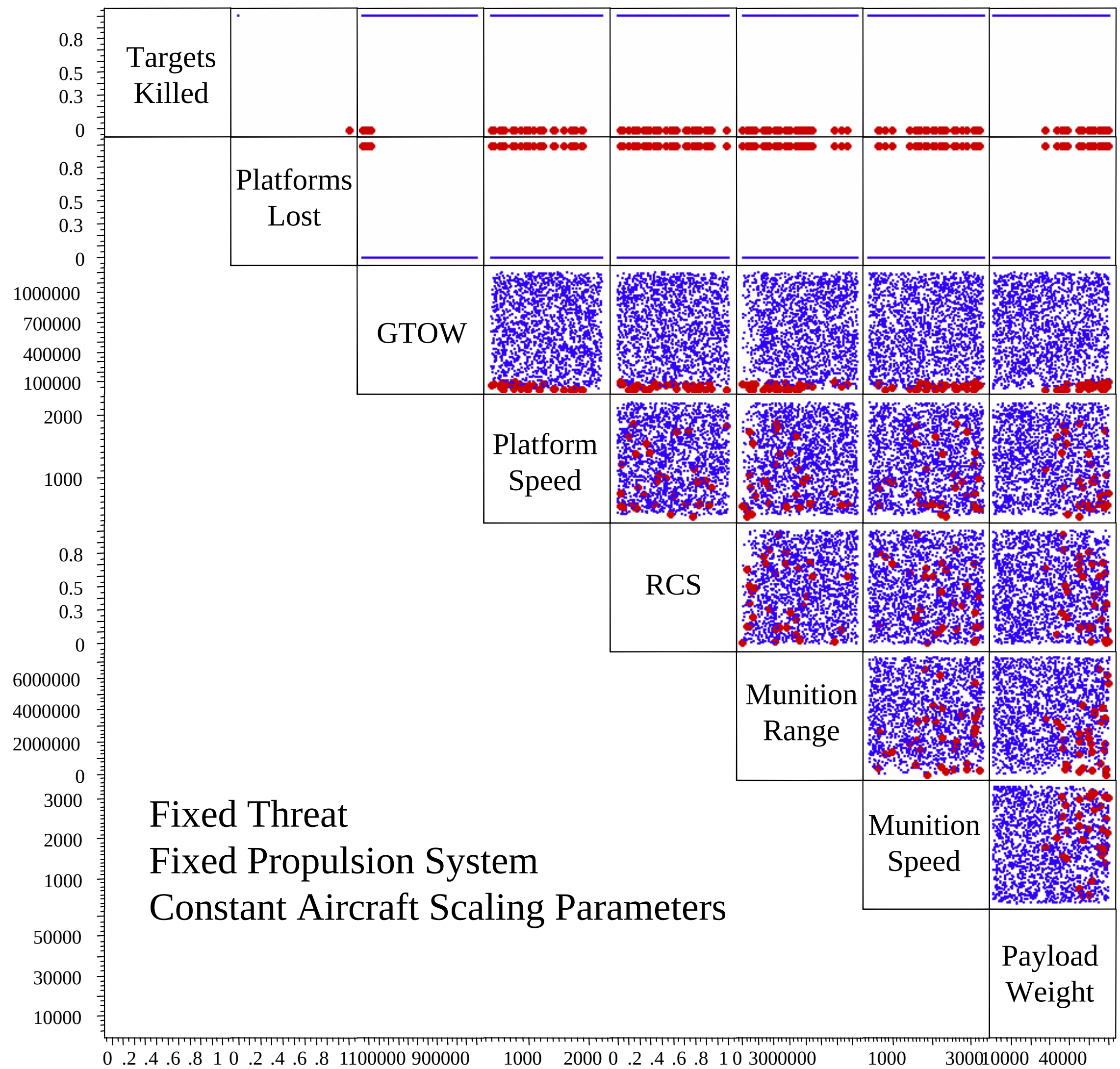


Figure 127: Multivariate Analysis for a Speed/Range/Stealth/Weight Trade Study.

128. The axes of the figure show the speed of the platform on the Y-axis and the radar cross section, a measure of stealth, on the X-axis. The upper left corner therefore indicates the fast and stealthy solution, and the lower right corner indicates a subsonic, non-stealthy configuration. When the blue points are examined, a clear Pareto frontier is visible between speed and stealth. The curve at the bottom of the blue region shows that for short range weapons, a slower platform requires a lower RCS for iso-survivability. If the platform speed is increased, the RCS can be increased commensurately along the Pareto frontier while still maintaining effectiveness. Essentially, a faster platform can be less stealthy and still

be successful. This Pareto frontier is not evident in the green or red cases that indicate successful engagements for standoff weapons. This is because only in the blue cases does the platform come close enough to the adversary to be detected. RCS therefore has a minimal impact on platform survivability when standoff weapons are used. The primary difference between the red and green areas are that the faster munition allows iso-effectiveness at a lower platform speed. In this scenario, platform speed and weapon speed can be traded equally for effectiveness. This is only true because the scenario is time-limited.

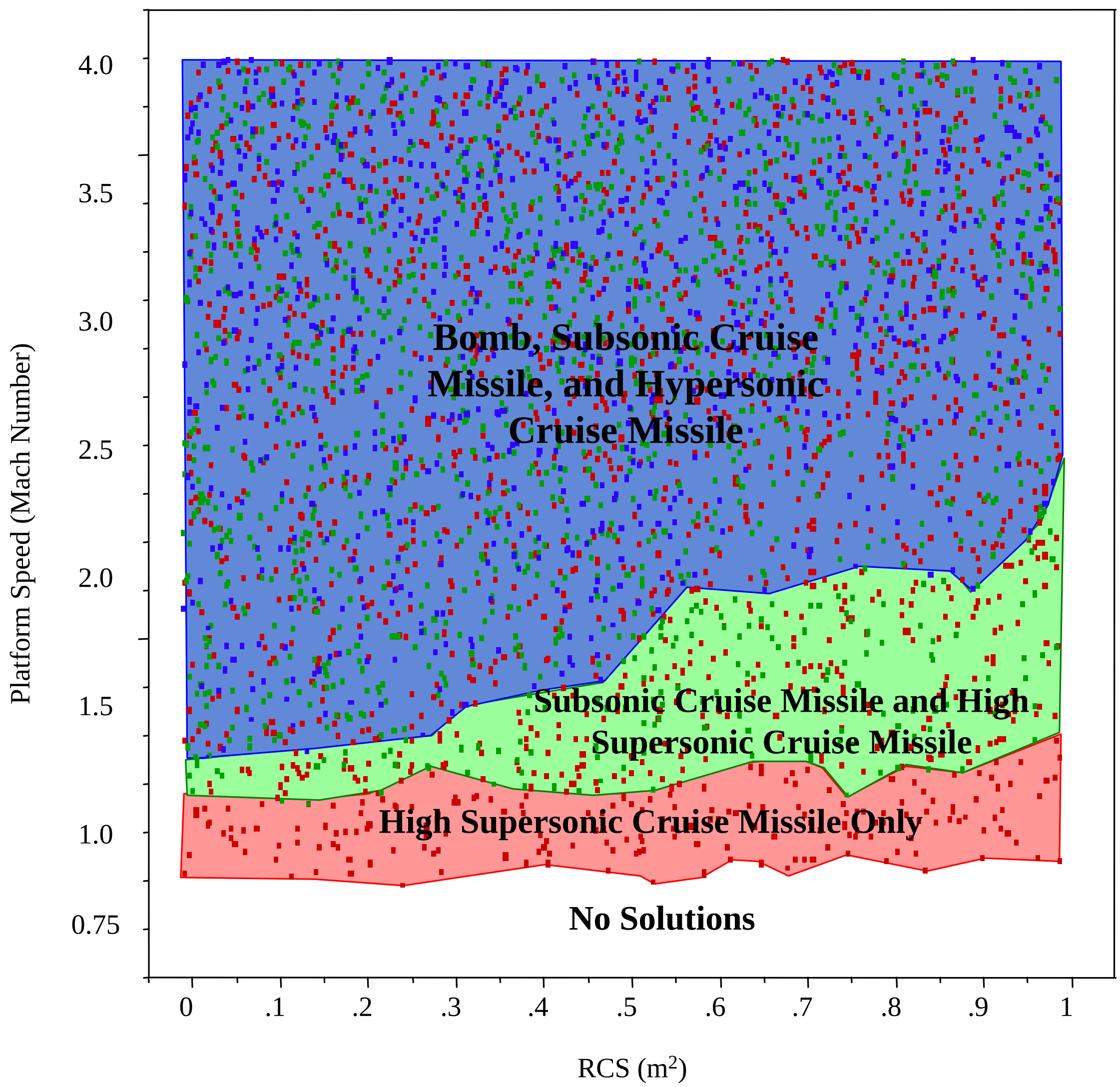


Figure 128: Tradeoff Between Platform Speed and Stealth for Three Fixed Missiles (Blue = Subsonic Bomb, Green = Subsonic Cruise Missile, Red = High Supersonic Cruise Missile).

This figure indicates that for this scenario, stealth is only a desirable characteristic when standoff weapons are not an option. While all three weapon combinations are successful in the blue shaded region, a military planner would tend to prefer less expensive bombs than long range standoff munitions in this case; however, if the platform speed and radar cross section cause the platform to be in the green or red regions, a more expensive munition is necessary to ensure platform survival.

Finally, it is important to note that these trends hold for the case defined in Table 31. The location of the "knee-in-the-curve" for the blue Pareto frontier changes as the assumptions are altered. The analysis of the data can be a cumbersome process because visualization tools do not yet exist to enable a dynamic tradeoff of assumptions within the multivariate scatterplot matrix.

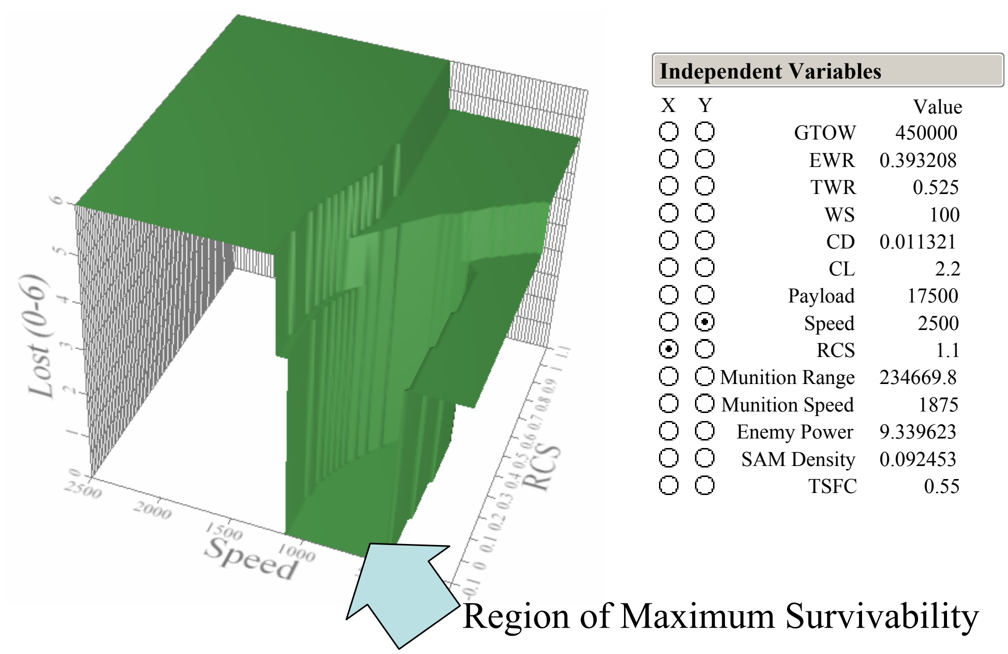


Figure 129: Three Dimensional Surface Profiler for Platform Speed and RCS for the GSTF Three Day Scenario.

The answer to this question changes somewhat as the parameters of the scenario and the design assumptions of the platform change. A three-dimensional tradeoff between platform speed and RCS with respect to platforms lost for the GSTF three day scenario is shown in Figure 129. Here here is a certain region where survivability is maximized. The exact values that bound this location are a function of the other design variables and the parameters of the threat: increasing enemy radar power or SAM density causes the space to form a plateau at the maximum level of platforms lost. For the scenarios examined, the LRS aircraft asset is very sensitive to an evolving threat.

5.10.4.6 Evaluation of Speed and Persistence

Both speed and persistence have been identified as desirable attributes for a future LRS system; however, do both of these factors need to be maximized to ensure mission success? At one end of the spectrum, a fast long-range cruise missile could be employed from any number of platforms. As mentioned in Section 5.7.4, the opposite end of the spectrum is an "area dominance" munition designed for maximum loiter time at low speed. Using the SOCRATES methodology, these two disparate concepts can be compared against the same top-level MoEs.

Four MoEs are tracked for the TCT attack mission: the percentage of TBM launchers killed, the number of TBM launchers killed, the number of TBMs fired, and the number of blue munitions fired. An analysis of these MoEs shows that they are very closely correlated.

To analyze the source of variability of the MoEs, the ANOVA procedure was performed for each of six mission variations for the TCT attack scenario. The number of cases for each of the subsequent variations was constrained by the number of runs that could be executed in 48 hours. Note that as more munitions are added, the number of cases decreases. This is because each area dominance munition has an onboard radar, and frequent track-quality radar calculations in FLAMES are computationally expensive.

1. Baseline Case (3200 cases, dark blue)
2. Area Dominance (3200 cases, red)
3. Area Dominance + Dispersed Formation (835 cases, green)
4. Double Munitions + Dispersed Formation (567 cases, pink)
5. Double Munitions (546 cases, orange)
6. Triple Munitions + Dispersed Formation (384 cases, light blue)

The Pareto chart, color coded for each tactical variation according to the above scheme, is shown in Figure 130. After the first two cases were executed, the TBM RCS was determined to have little effect on the detectability of the target and was removed from subsequent tactical variations to reduce the number of regression variables by one.

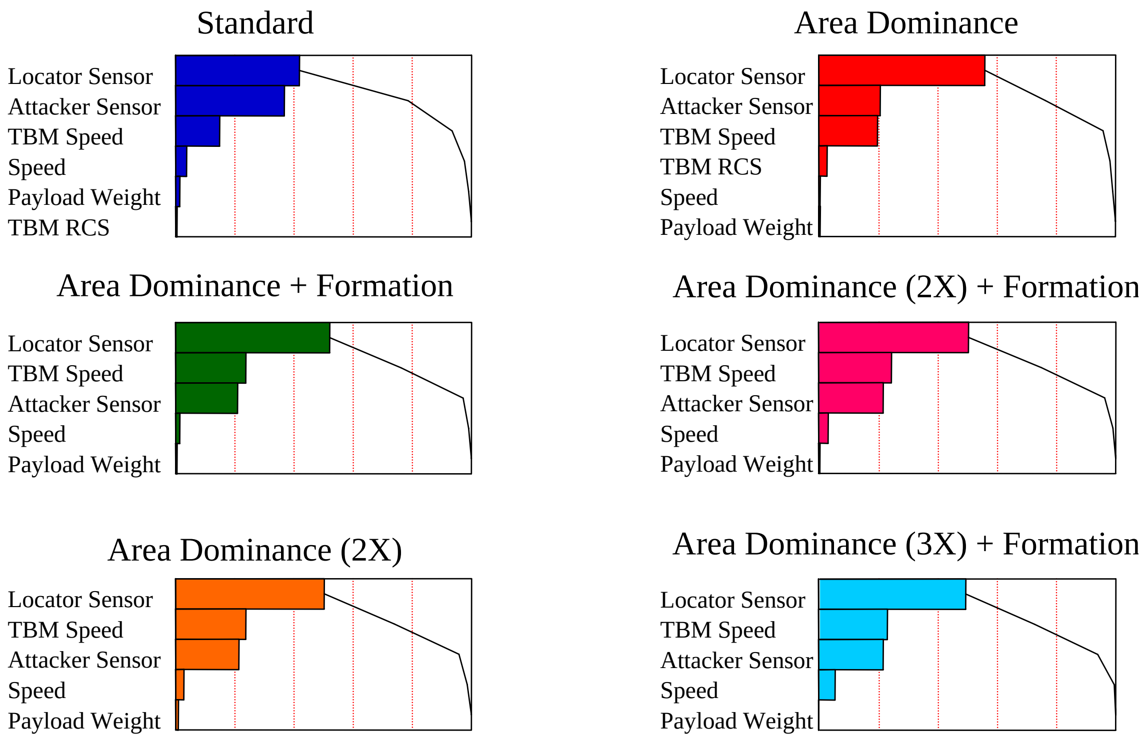


Figure 130: Pareto Chart for the TCT Attack Scenario Divided by Tactical Variations.

For the baseline case, the impact of the fighter's sensor is greater than that of the area dominance cases. In general, the speed of the TBM and the range of the onboard sensor are the next most dominant parameters after the airborne sensor's range. The speed of the platform/munition and the payload weight (which defines the number of submunitions carried) is in the noise by comparison.

The data generated from the simulation runs for each of the aforementioned variations is shown using multivariate analysis in Figure 131.

This figure shows three MoEs, the number of TBMs killed, the number of friendly munitions fired, and the number of hostile TBMs fired as the key tracked metrics in the upper left corner. The next two parameters relate to the speed of the platform/weapon

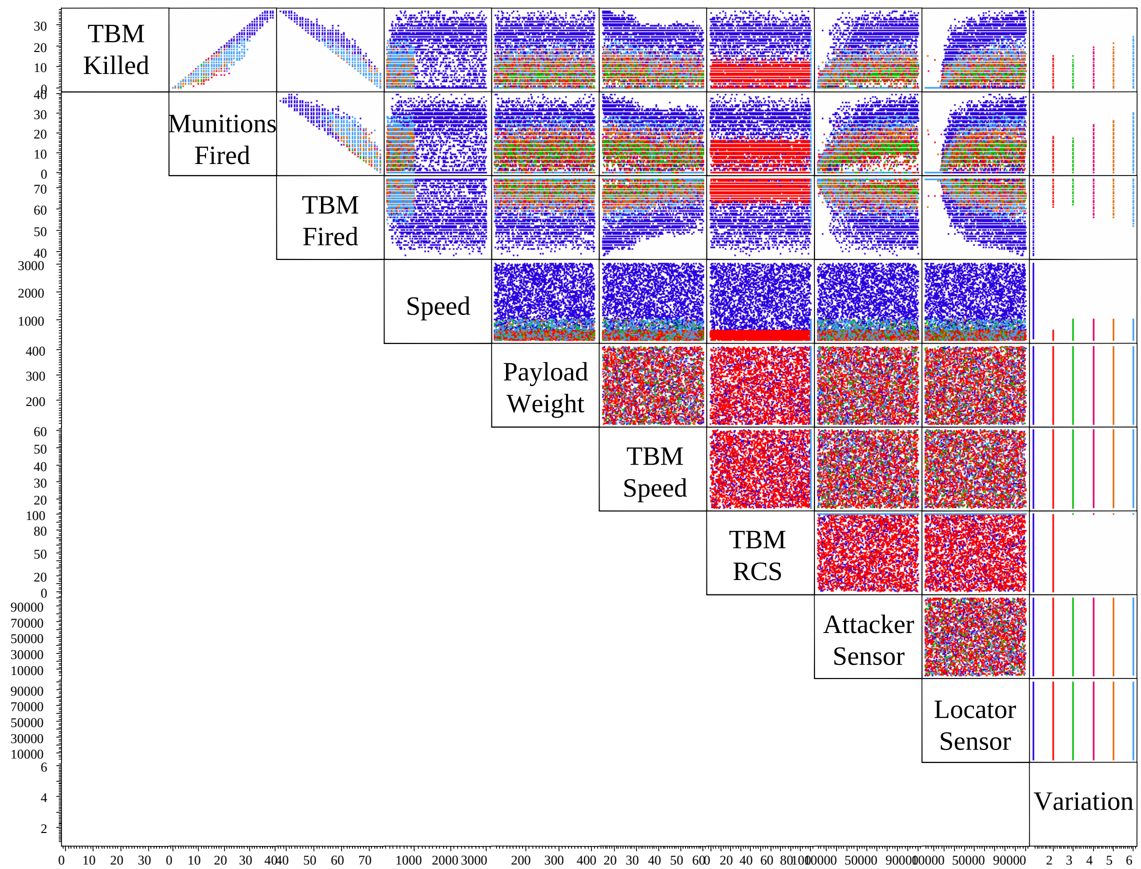


Figure 131: Multivariate Analysis for the Time Critical Strike Scenario.

and the number of munitions/submunitions it can carry. The next two parameters define the hiding characteristics of the enemy and the next two parameters define the sensor subsystems of the attacker and locator respectively. The final parameter is a switch that represents the color-coded tactical variation according to the numbering scheme mentioned above. As an example of how this data can be used for capability analysis, the filtered Monte Carlo technique can be applied to this multivariate analysis to identify situations where more than thirty TBM launcher are killed as shown in Figure 132. As illustrated in the figure, only fighters with high speed missiles are capable of reaching this threshold value for the TBM killed MoE.

The same set of data can be further used to analyze the "best-in-class" solutions for each of the six tactical variations. Figure 133 shows a zoomed-in version of the upper-right relation in Figure 131. This figure depicts the number of TBMs killed for each tactical

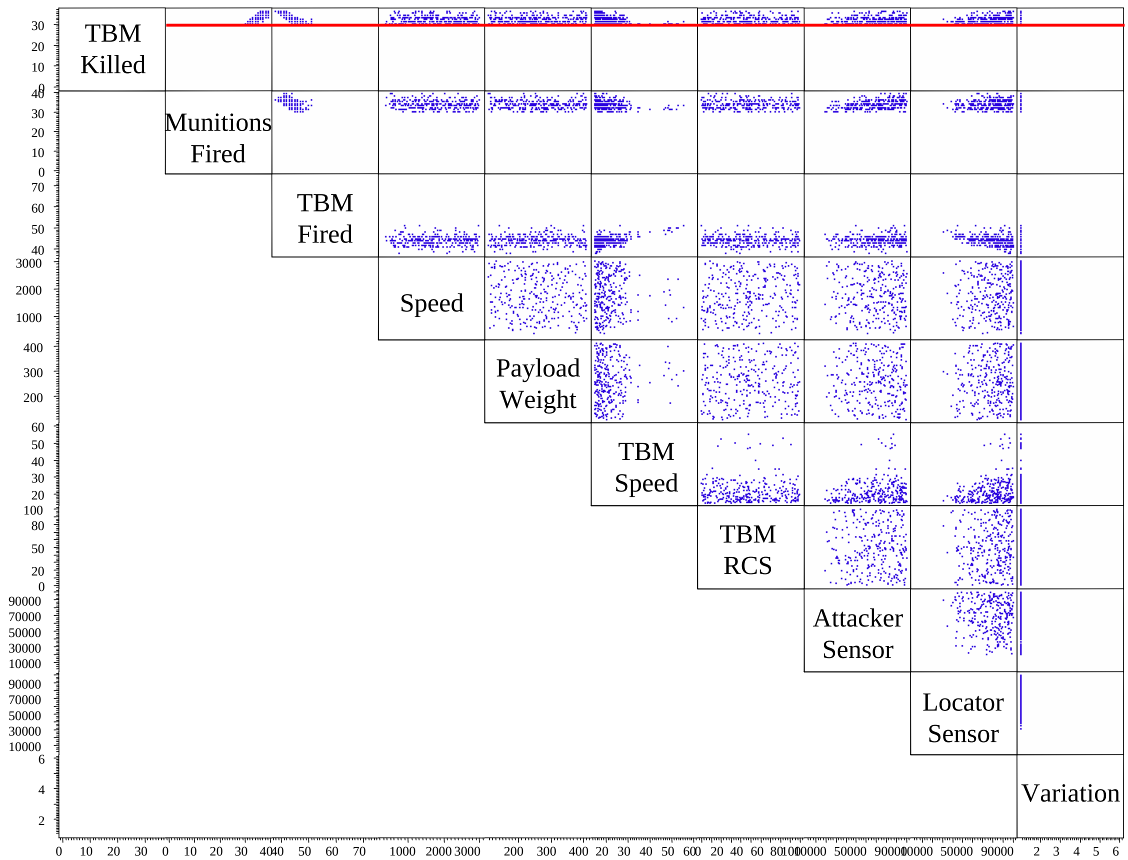


Figure 132: Time Critical Attack Designs That Kill More Than 30 TBMs.

variation. The top solutions in each column are highlighted and the remaining cases are excluded to produce the multivariate plot in Figure 134 that depicts the top solutions for each variation. As shown by the insets in the lower left of the figure, there are thresholds of effectiveness for each of the color coded variants. Also, the population of "best-in-class" solutions tends to be located in the region with both sensor ranges maximized.

At first glance, it also appears that the best area dominance munitions are capable of attacking faster moving TBMs, however, unhiding the blue points reveals that they do universally better in all dimensions. An analysis of the Pareto frontier between TBM speed and TBM killed reveals that the non-dominated solutions are those with double and triple munitions. The final example analysis performed on the TCT attack scenario examines the impact of munition type and the use of dispersed formations. A series of Pareto frontiers are highlighted in Figure 135 for the tactical variations previously discussed. The shaded

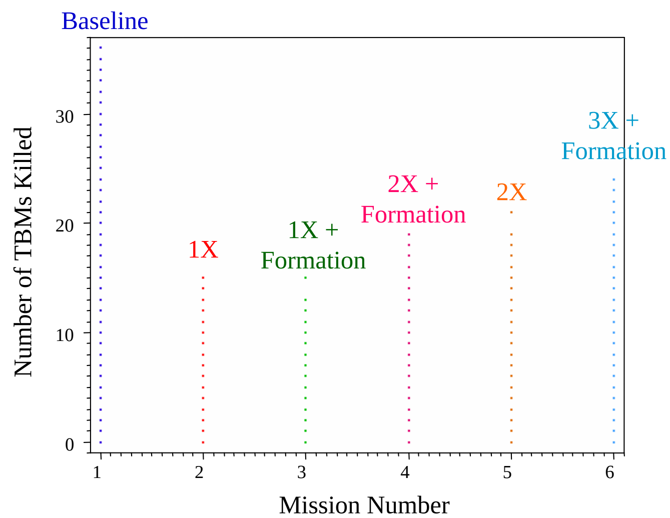


Figure 133: Analysis of TBMs Killed for Each of the Six Tactical Variations.

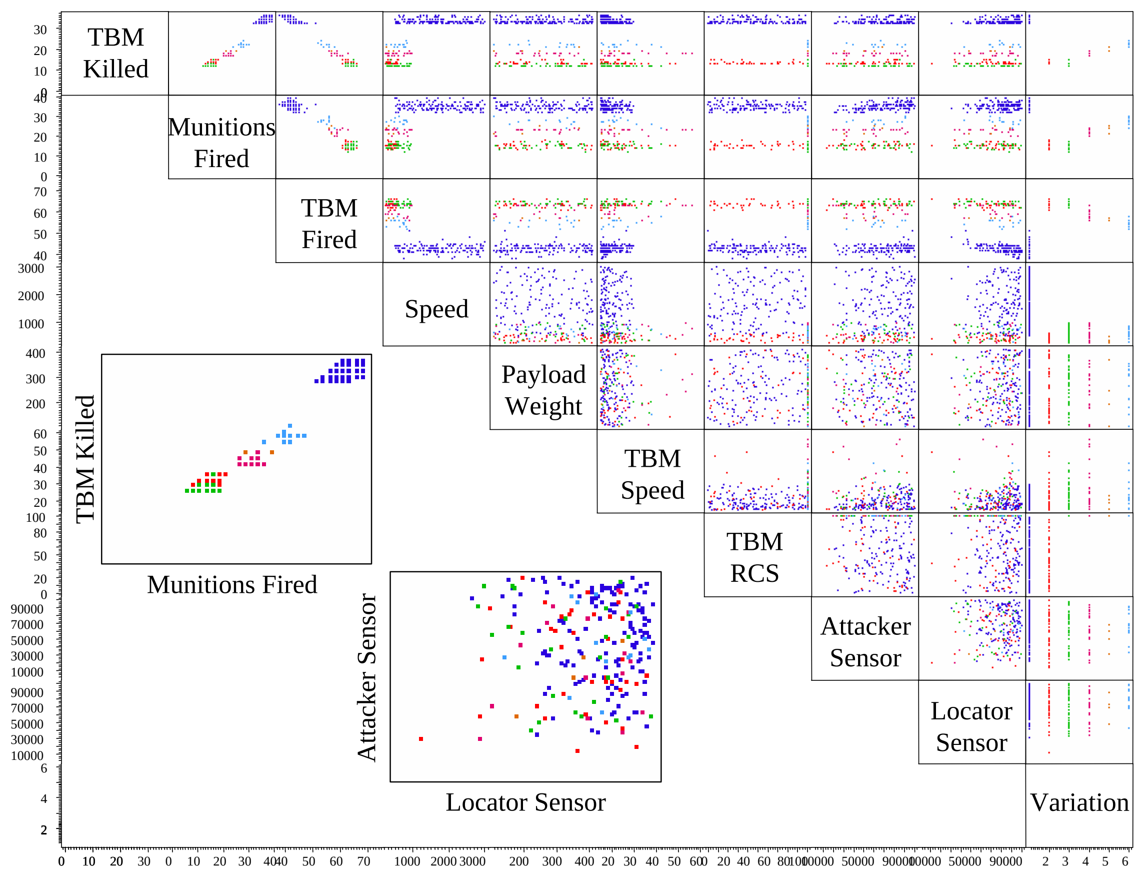


Figure 134: Analysis of "Best-in-Class" Solutions for TCT Attack.

area represents the solution region encapsulated by area dominance solutions. The colored lines are the thresholds for each variation respectively. Surprisingly, all evidence supports the notion that the dispersed munition formation is *less* effective than when all munitions loiter near the airborne sensor. This must indicate that the likelihood of reaching a target is increased when the munition starts near the sensor that originally located the target, implying that for this scenario, directly launching targeted weapons may be more effective than a netted force of geographically distributed weapons. Further development of cognition algorithms for the battle management of area dominance munitions is needed to further explore this phenomenon.

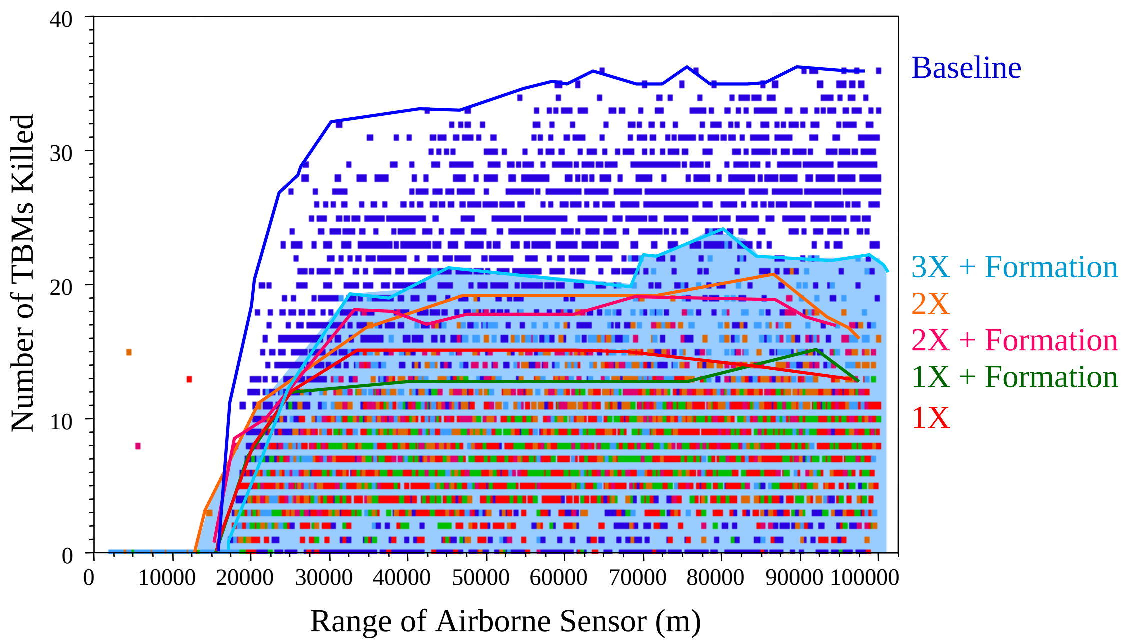


Figure 135: Pareto Frontiers for TCT Attack Effectiveness.

5.10.4.7 *Evaluation of Evolving Threats*

Many entry-into-service dates from 2018 to 2037 have been proposed for a future LRS system. Implicit in these definitions is the anticipated threat that an LRS system architecture must be effective against. Since these threats evolve rapidly over time, a "point solution" at one of these arbitrary dates is not appropriate. One of the key aspects of the results from the SOCRATES methodology is a parametric exploration of adversary characteristics in conjunction with blue force assumptions, tactics, designs, and technologies. A simplified depiction of this tradeoff is shown in Figure 136 for two different parametric aircraft operating in the same scenario. The contour profiler depicts contours of platforms lost (green) and an arbitrarily selected threshold of 70 targets killed (blue region). The axes represent the two adversary parameters of the predictive neural net equation for the GSTF three day scenario: SAM density and the enemy radar power. As the figure illustrates, there is a small feasible region at low SAM density and low enemy radar power. Changing the input variables to the neural network as shown in the right side of Figure 136 causes little change in the platforms lost contours but opens up a region of effectiveness where the platform can eliminate more targets below the SAM density threshold of 20%. Unfortunately, no combination of the input variables significantly increases effectiveness as the threat density and enemy radar power increase.

Since it is difficult to demonstrate the dynamic nature of the contour profiler shown in Figure 136, a three-dimensional view that depicts the two threat axes with respect to the targets killed MoE is shown in Figure 137. In this case, the far corner represents high SAM density and high enemy radar power. This surface is also parametrically variable with respect to platform and munition design variables. Increasing the weapon range, for example, dramatically shifts the surface upward. There is a preferred bound on the platform speed around Mach 1.5. Very low speeds are inefficient for prosecuting large numbers of targets within the 24 hour time window while very high speeds use excessive amounts of fuel and limit the effectiveness of individual sorties due to the need to frequently return to base and refuel.

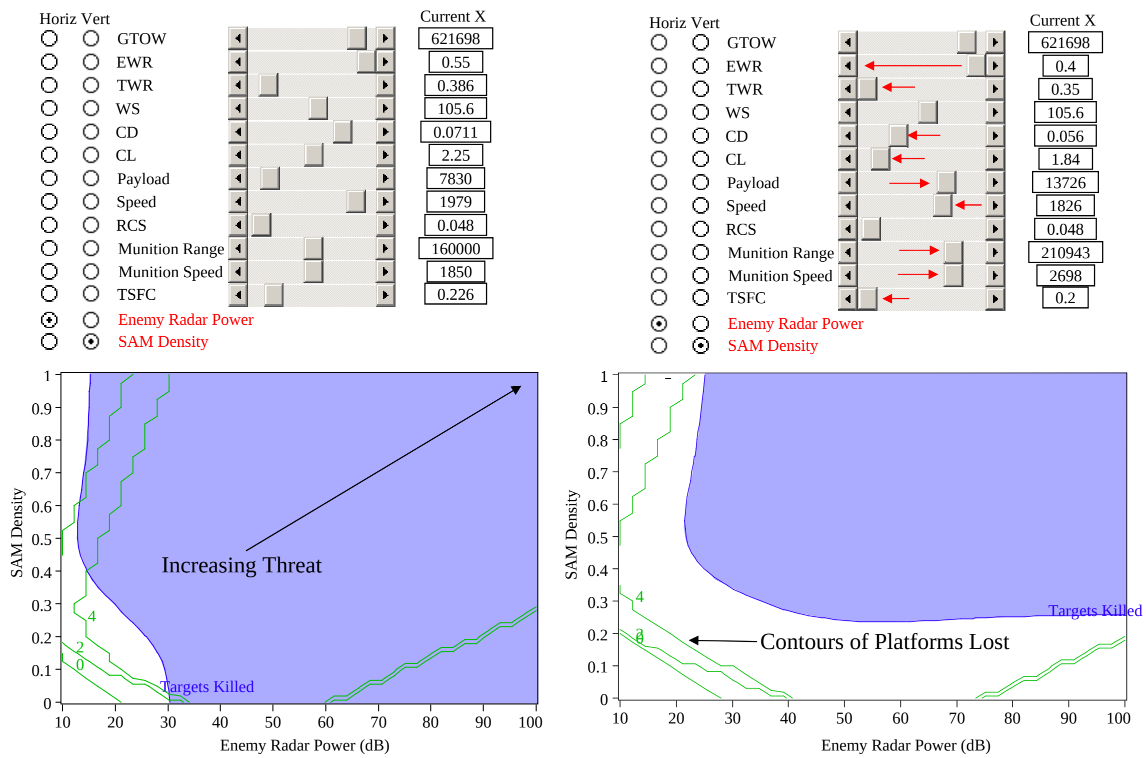


Figure 136: Contour Profiler for SAM Density and Enemy Radar Power.

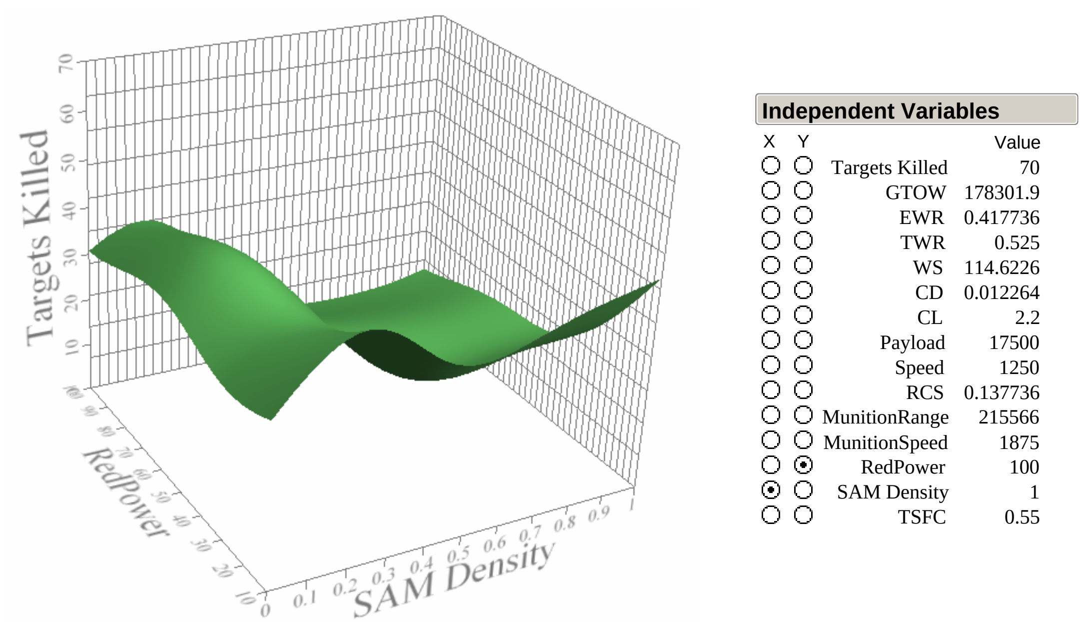


Figure 137: Three Dimensional Surface Profiler for SAM Density and Enemy Radar Power.

5.10.5 Comparison of Technology and Numbers

To play the devil's advocate, what if one took a page out of the Soviet playbook and simply replaced advanced technology solution with greater numbers of low-tech solutions? The baseline scenario used six B-2A-based platforms. To assess the sensitivity to changing the number of platforms, simulations with variations of 7, 8, 9, 10, 11, 12, 14, 15, 18, 20, 25, 30 and 40 platforms were executed. The results for the MoEs of targets killed and platforms lost are shown in Figures 138 and 139 respectively. The best technology-infused six platform case resulted in the neutralization of 263 targets. Surprisingly, this threshold was exceeded when over sixteen standard platforms were used. As shown in Figure 138, more than 25 B-2A aircraft are capable of prosecuting all targets in the scenario[30].

The platforms lost response shown in Figure 139 provides additional insight into the complexity of the problem. As more aircraft are added to the simulation, the *percentage* of the total platforms lost decreases exponentially. This refers back to the "wingman phenomenon" observed in Section 5.10.4.1. The very presence of more platforms means that the defending SAM sites must divide their fire among the observed threats instead of concentrating firepower against a single adversary. This implies that, at the very least, a strike package should be accompanied by drones or other aircraft that saturate the defender's view of the battlefield.

This example demonstrates how numbers and technology are interchangeable to a degree; however, the number of platforms must be increased by 166% to achieve parity with the best technology-enabled solutions. Other dimensions such as logistics, training, and maintenance are needed to completely compare these two classes of solutions. Finally, it is important to note that for the relatively benign scenario used, even if all sixteen combat coded B-2A bombers were used, more than half would be lost in the first 24 hours of combat. This fact alone is one of the key drivers for the pursuit of highly effective advanced technology solutions.

[30] Based on the assumption used in the baseline scenario and with a munition P_{Kill} of 100%.

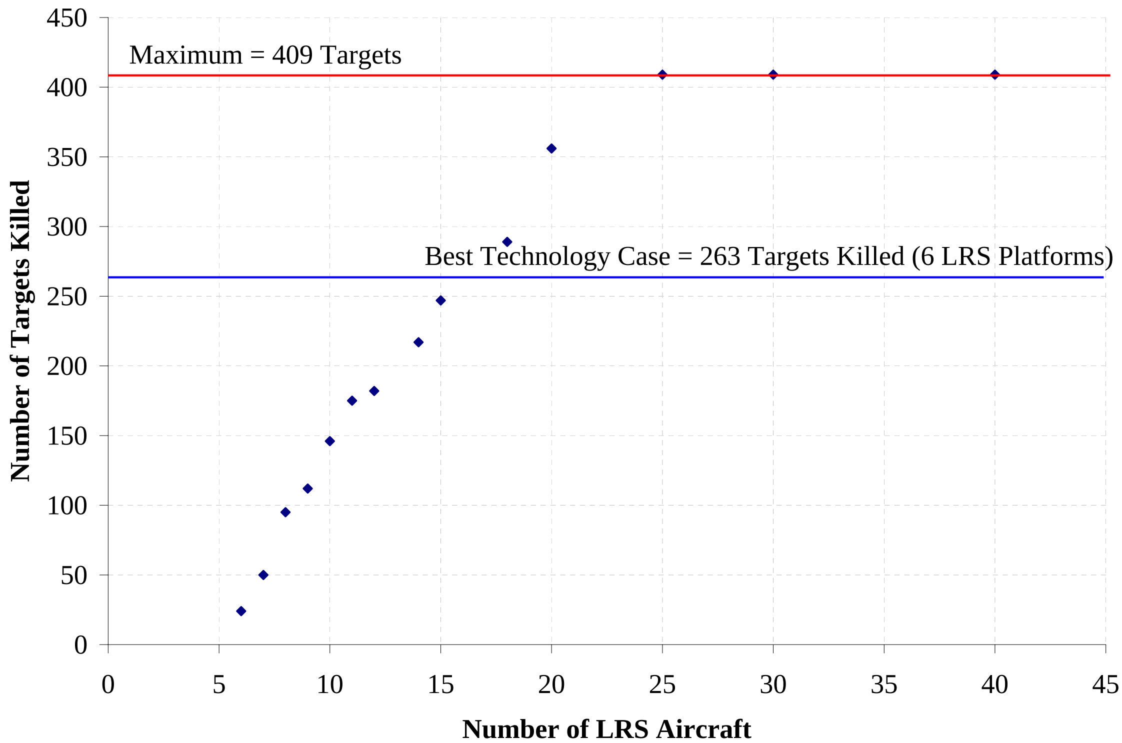


Figure 138: Sensitivity of the Targets Killed Response to the Addition of More Platforms.

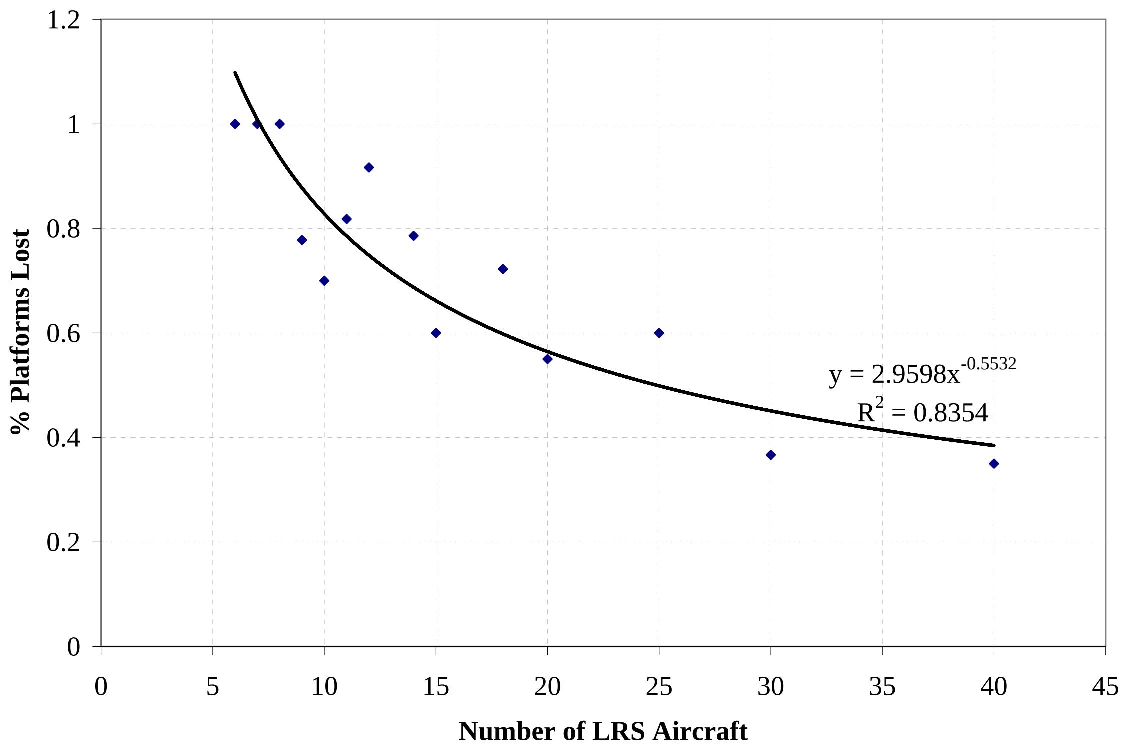


Figure 139: Sensitivity of the Platforms Lost Response to the Addition of More Platforms.

5.10.6 Quantifying Uncertainty

"There are known unknowns. That is to say we know there are some things we do not know. But there are also unknown unknowns, the ones we don't know we don't know."

-Donald Rumsfeld

While definitions of uncertainty abound in the literature, it is literally, the presence of something we do not know or cannot quantify. The previous analyses were performed in a *deterministic* manner, that is, a single input produces a single output with absolute certainty. Although the demonstrated method quantitatively traces technology impacts through multiple levels in a system-of-systems, as with all forecasting activities, the initial establishment of technology impacts is an imprecise art. Probabilistic techniques have emerged as a popular means to quantify uncertainty because of their statistical validity and ease of use. For example, Kirby utilized probabilistic techniques and Soban implemented a probabilistic system-of-systems effectiveness methodology to address this issue [240, 378].

Monte Carlo simulations, summarized in Section C.11, can be can be used to obtain probabilistic results from deterministic tools. Because they typically require thousands of runs their direct implementation on a physics-based code may not be feasible or practical. On the other hand, surrogate models run very quickly. The surrogate models created in Step 9 of the SOCRATES method are therefore ideally suited to implement probabilistic analysis of technology benefit. Quantification of uncertainty using probabilistic techniques and its impact on capability-based technology evaluation is addressed only briefly in this section. The recommendations section identifies a need for additional research in this area.

The first observation regarding the technology uncertainty is that precise statements regarding capability satisfaction cannot be made when uncertainty is present. This is illustrated in Figure 140.

In the left side of the figure, four discrete solutions are depicted against axes of effectiveness and cost. The ideal solution is therefore in the upper left corner where effectiveness is maximized and cost is minimized. In a deterministic analysis, point 1 is the closest to the

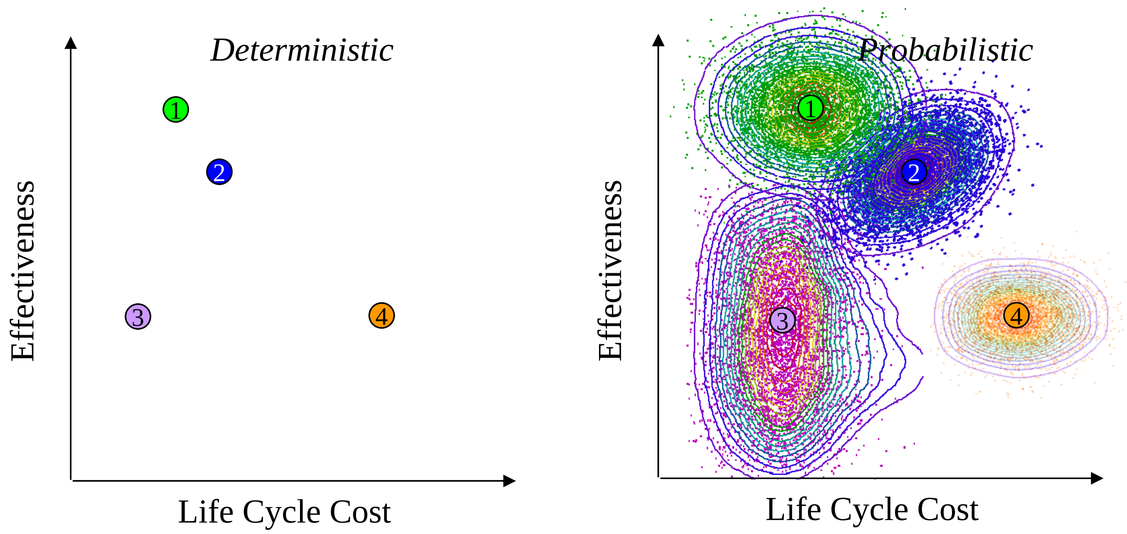


Figure 140: Notional Depiction of Four Points without Uncertainty (Left) and with Uncertainty (right).

optimum solution, point 3 is the least expensive, and point 4 is clearly the worst. When probability distributions are added and their effect is observed as shown in the right side of Figure 140, the "answer" is less clear because of the spread of the distributions around the deterministic value. In the probabilistic view, points 1, 2, and 3 have distributions that overlap. Therefore any of the three points could be located in the overlapping region with a finite probability. In a probabilistic sense, these points are now indistinguishable although their probability of overlap can be calculated. Point 3, which previously would have been eliminated due to low effectiveness, now has a wide range of potential effectiveness. On the other hand, Point 4 has a relatively small distribution that does not intersect the other three. The only conclusion that can be reached in this analysis is that Point 4 is still the least ideal solution.

This analysis can be extended to the observation of Pareto frontiers in the effectiveness vs. cost space as shown in Figure 141. The locus of non-dominated solutions shown by the blue points on the left side of Figure 141 form a clear Pareto frontier. The solutions along this frontier form a pool from which an "optimum" can be selected depending on how the customer chooses to weight the two axes.

Unfortunately, when probabilistic analysis is coupled to the calculation of the Pareto

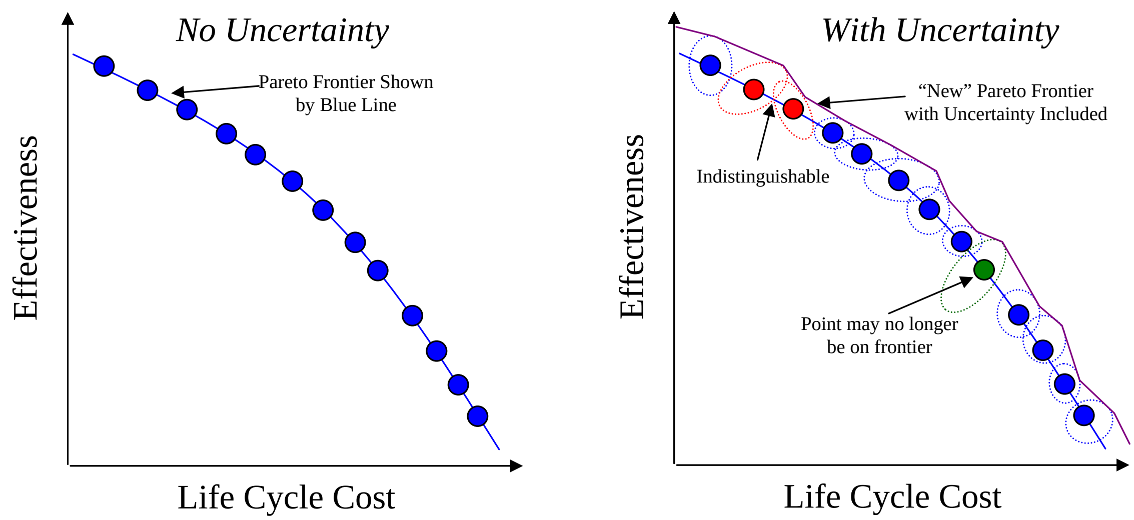


Figure 141: Analysis of Probabilistic Pareto Frontiers.

frontier, the clear delineations disappear as shown on the right side of Figure 141. For example, the two red points have overlapping distributions, meaning that either point could be at the other's location. The uncertainty distribution around the dark green point extends deep inside the region of dominated solutions. This means that there is a finite probability that this point is no longer along the frontier and a previously dominated solution may have taken its place. The construction of a line tangent to the probability distributions of the outermost points (purple line) forms a new Pareto frontier, which indicates the absolute threshold of effectiveness and cost with uncertainty included. It is also important to note that the true boundary could be located inside the blue line when uncertainty is evaluated.

This conceptual analysis is demonstrated using the surrogate models developed in Step 9 of the testbed demonstration in Figure 142. This figure builds upon the deterministic analysis shown in Figure 108 for the GSTF three day scenario. The upper left corner of the figure shows the relationship between targets killed and platform speed when speed and TSFC are varied uniformly across the allowable range. The other three insets in the figure show how this distribution of points changes and becomes more "fuzzy" as random noise is added to the targets killed response. Normal distributions with a standard deviation of 1, 2, and 3 are shown for illustration. The blue points are the 1,000 original baseline points and the black points indicate 100 Monte Carlo runs added to each of the baseline cases.

As the figure shows, the boundary between the two variables becomes much less defined. In the deterministic case, the "best-in-class" solution occurs at a speed of about 700 knots and 30 targets killed. When uncertainty is added, the mean speed value shifts somewhat, but the targets killed metric takes on values up to 33% greater.

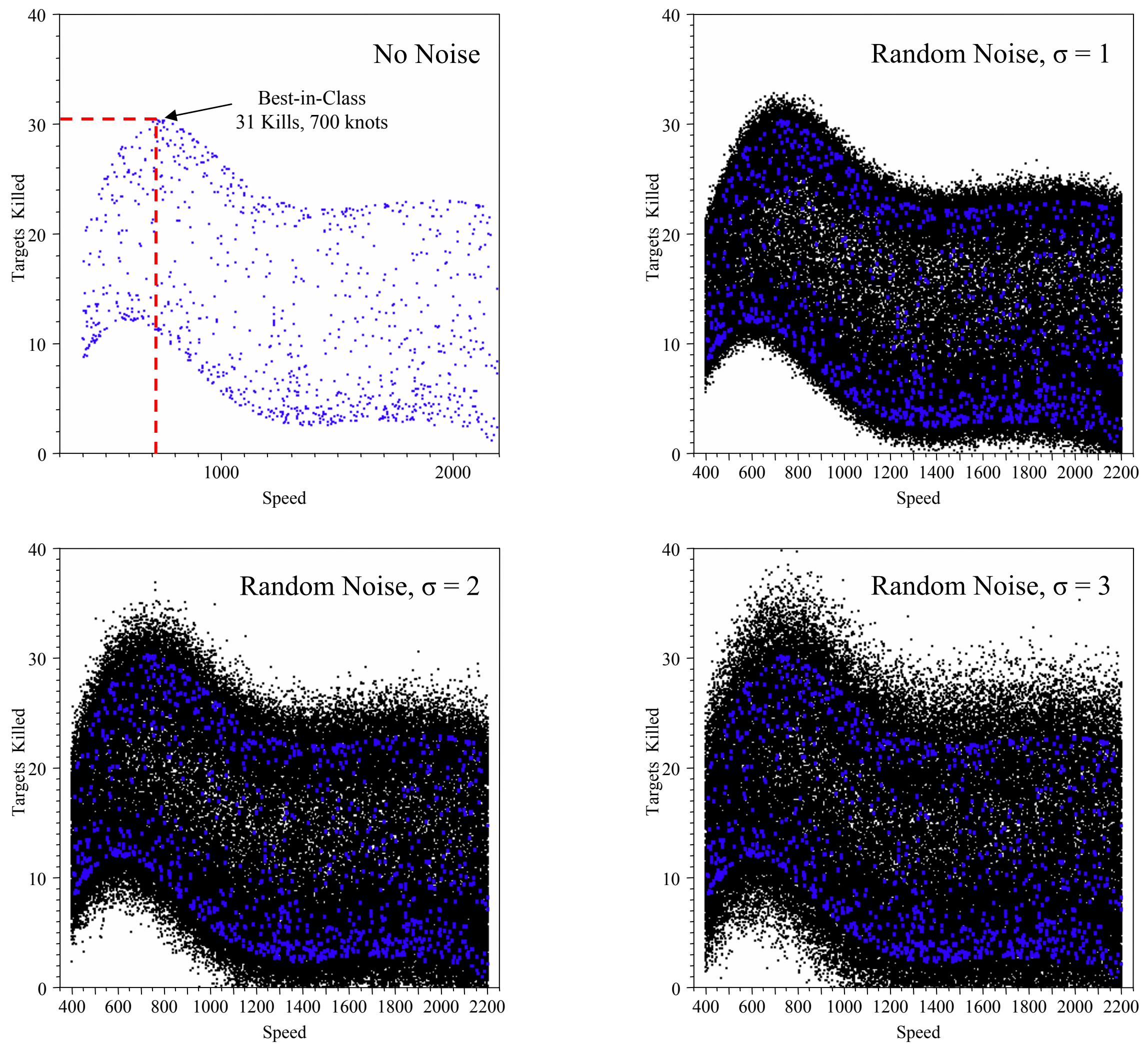


Figure 142: Addition of Random Noise to a Design Space Exploration of Speed and TSFC.

In addition to applying noise distributions to the output variables, uncertainty distributions can be applied to the input variables. Using an example from the TCT attack scenario, an application of normal distributions to three input variables is shown in Figure 143. The variation in the sensor ranges and TBM speed can, for example, simulate different types of weather that impede visibility and mobility. In the deterministic case, 23

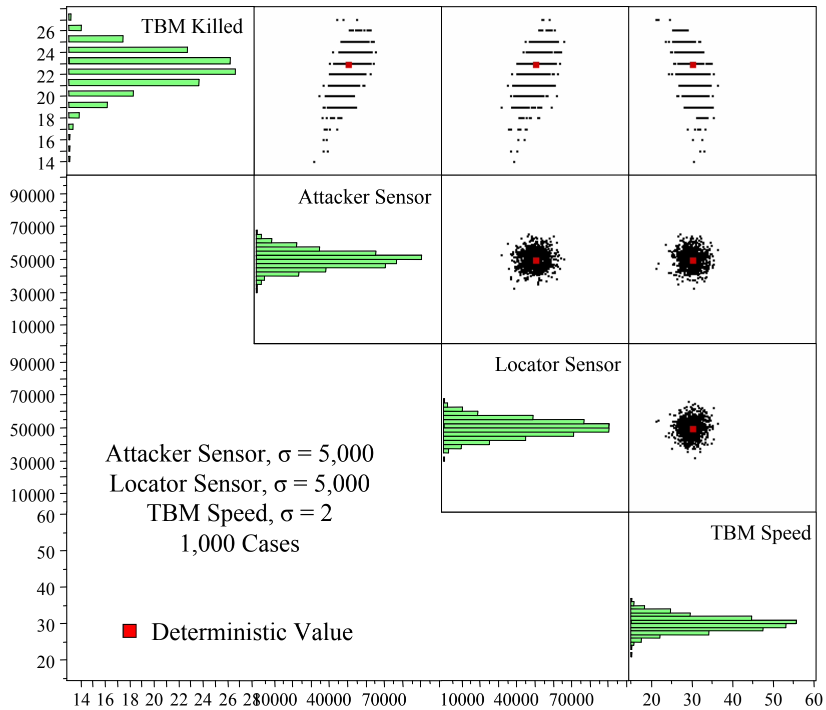


Figure 143: Application of Uncertainty Distributions to Input Variables.

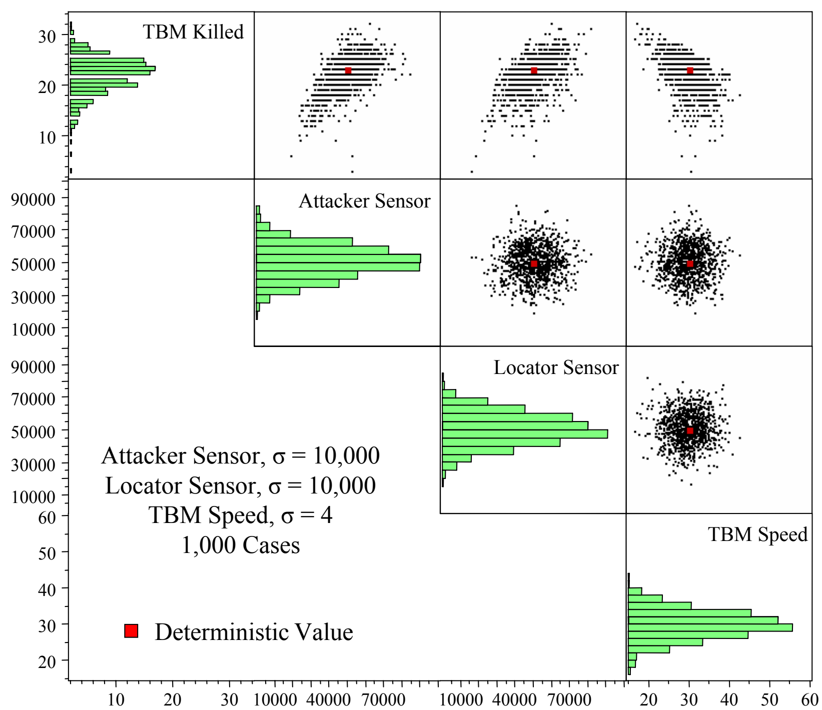


Figure 144: Application of Larger Uncertainty Distributions to Input Variables.

TBM launchers are killed by this aircraft configuration. Using a normal distribution with standard deviations as labeled in Figure 143, the expected number of TBMs killed ranges from 14 to 27 and the top row of metrics take on a characteristic teardrop shape. When the spread of the distribution is increased as shown in Figure 144, the range of expected TBMs killed ranges from 6 to 32. Figure 145 shows the combination of uncertainty distributions on the input variables combined with a 100 case Monte Carlo distribution on the output parameters. The blue line in this figure highlights the boundaries of the distribution as drawn in Figure 144 using fewer cases.

When uncertainty distributions are included on either the input or output parameters, it quickly becomes difficult to differentiate between variation caused by uncertainty and variations caused by a change in one or more design variables at one or more hierarchical levels. Further research is needed to address this challenge.

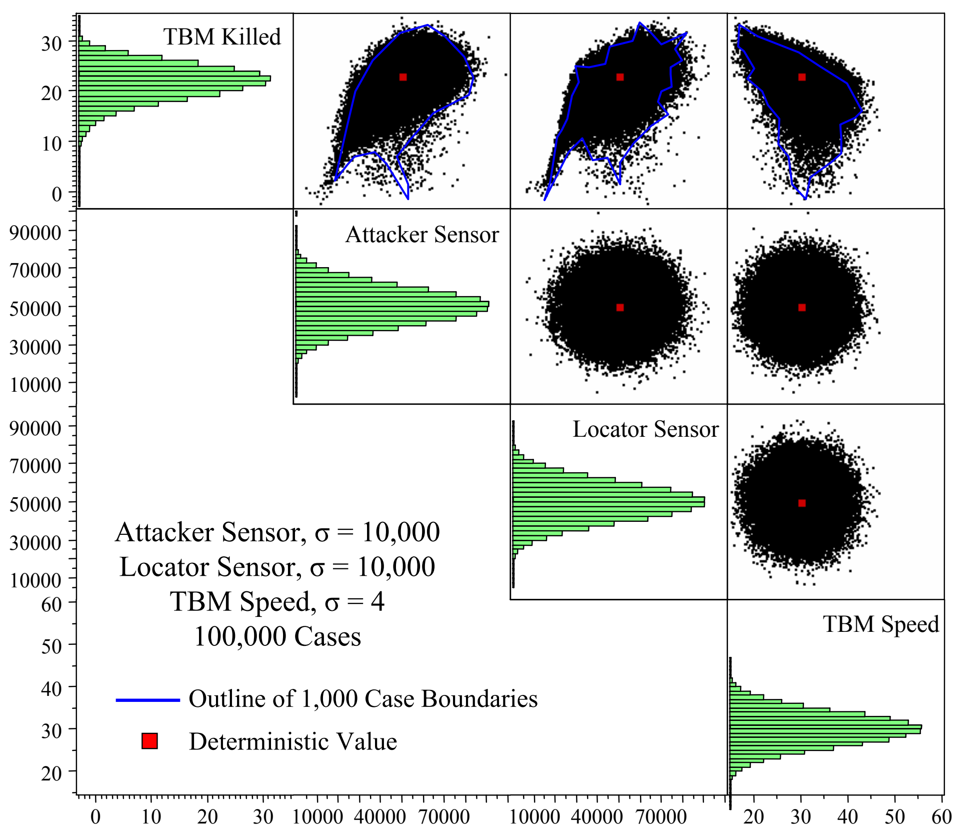


Figure 145: Application of Uncertainty Distributions to Input and Output Variables.

5.10.7 Using The FLASH Playback Files to Understand Agent Decisions

The FLAMES Scenario Highlighter, or FLASH, is a component of the framework that enables playback of recorded scenario files. These two-dimensional and three-dimensional animations of the simulation results act as a "God's Eye" view of the conflict. Although such results are sometimes coupled with imagery and high-fidelity terrain data to produce digital eye candy and high-profile special effects, the playback files also allow a user to debug agent behaviors and discover tactical decisions being made by the agents. Two playback files can also be executed side-by-side to view the differences between two simulations. In this manner, a designer can see where the agents digress from a certain course of action as a function of the technologies they are provided. Although difficult to depict on paper, the baseline FLASH scenario file was often compared to DoE results throughout this work to debug the scenario and identify unanticipated behavior.

By saving the playback files associated with each DoE run, after narrowing the design space to a region of interest, the designer can review the playback files *in that region.* Since each file takes several minutes to view, it is not practical to watch every playback file. The multivariate filtering technique can be used to narrow down the number of scenario playback files to be viewed.

An example of a FLASH playback file is shown in Figure 146. This example shows thirty B-2A platforms attacking a series of targets in Iraq. The group of platforms flies out together in four small clusters and "breaks" in a starburst pattern after reaching the end of the first corridor. This screenshot is taken approximately 2 hours into the scenario. Here, one platform has been lost in the 4th Air Defense Sector in northern Iraq. Another platform is shown firing an air-to-ground munition and egressing back towards an airspace corridor. In this playback file, agents follow the corridors, release munitions, and return to base as programmed.

A later screenshot of the same playback file is shown in Figure 147. Here, many platforms are shown egressing through Path 11, a popular low-threat route. This route is not intersected by any SAM sites and is also one of the shortest routes to attack targets in downtown Baghdad. The green color-coded platforms have expended all munitions and are

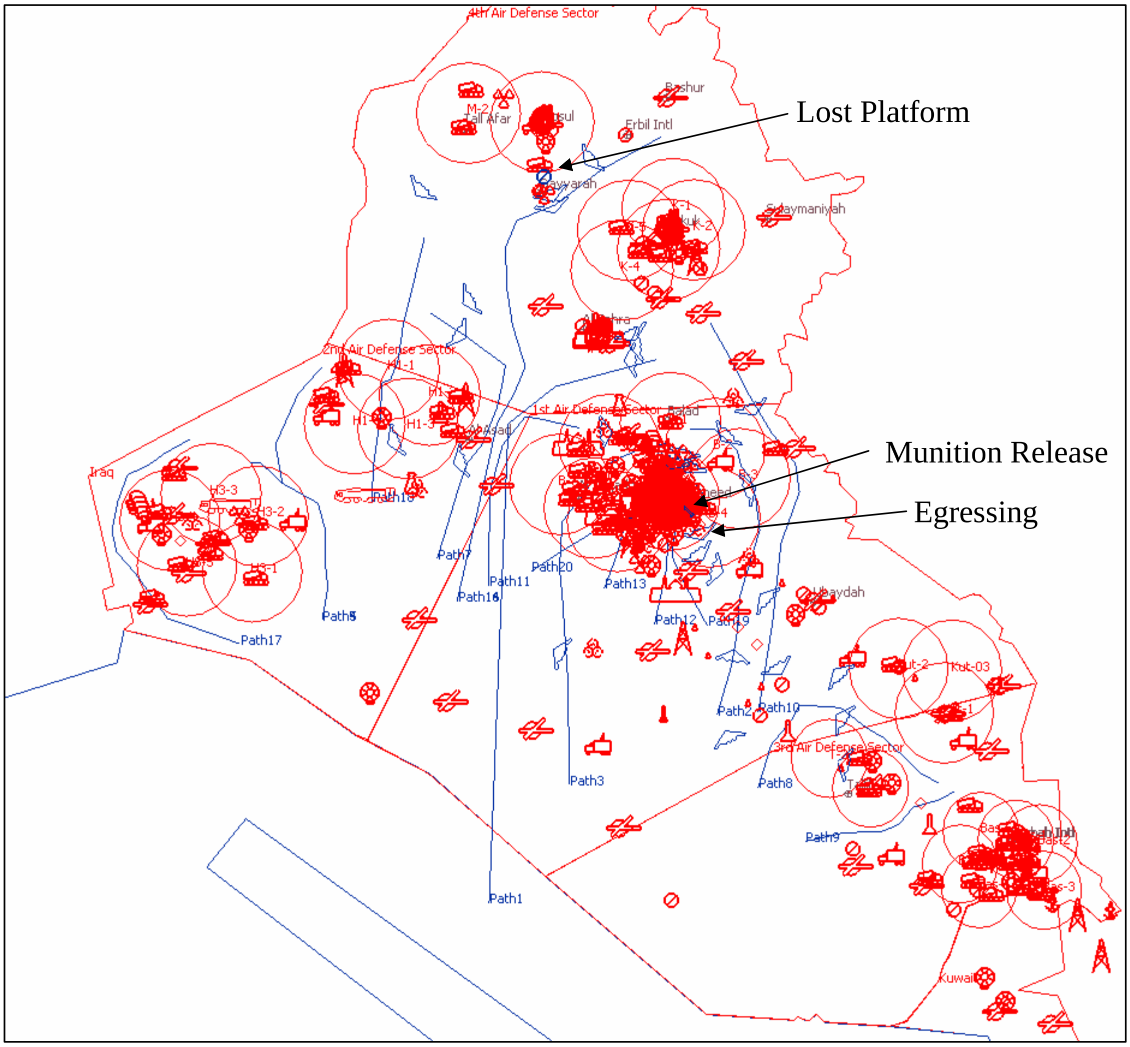


Figure 146: FLASH Scenario Playback File.

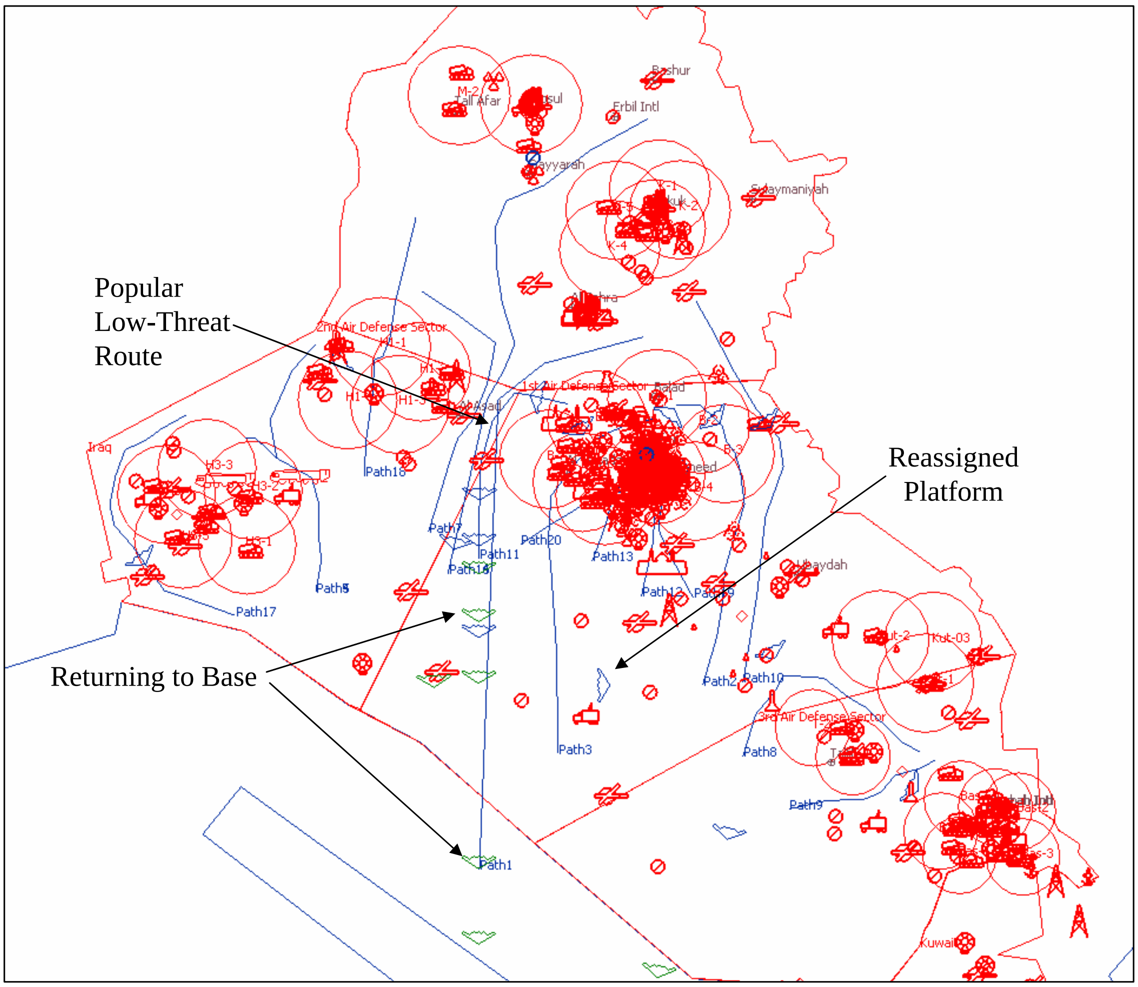


Figure 147: FLASH Scenario Playback File Showing Route Selection.

returning to their Saudi base to rearm. The blue platforms interspersed between them are about to be reassigned to other targets in Eastern Iraq or in Baghdad. Here, the designer is trying to ensure that the platforms make logical selections in their choices of airspace corridors. This is one of the only ways to check the validity of the objective function used for route selection.

Zooming into the playback file at a later time (Figure 148) shows the agent behavior around the heavily defended portion of downtown Baghdad. Here a variety of behaviors are observed.

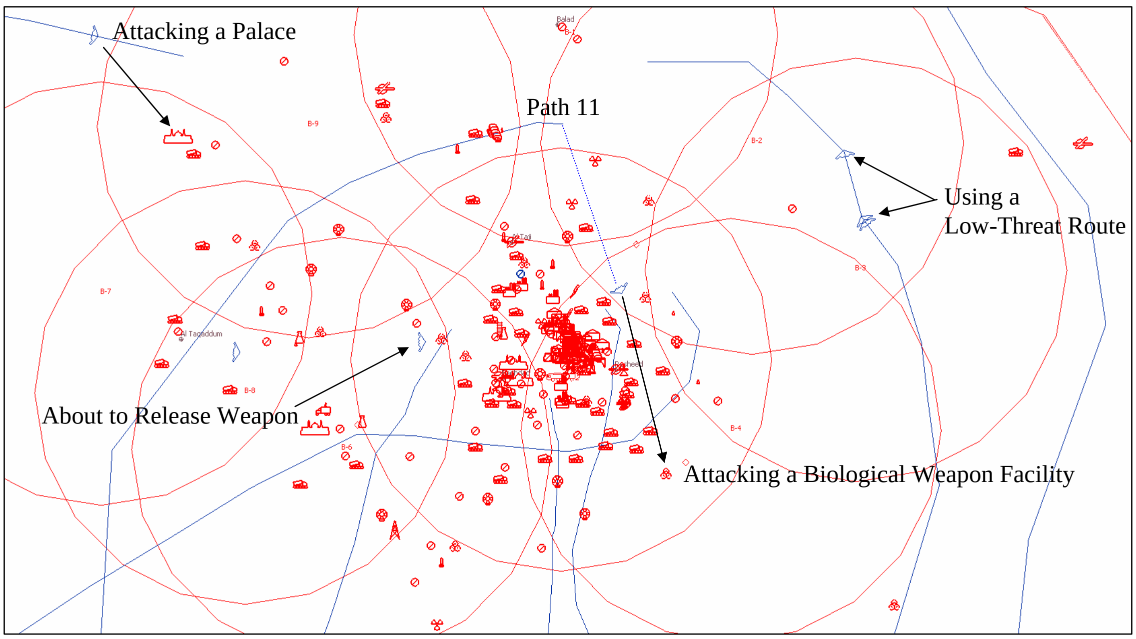


Figure 148: FLASH Scenario Playback File Showing Route Selection and Aggressor Behavior.

First, in the upper left corner, a B-2A bomber is flying along a corridor and preparing to attack a palace in the northwest corner of Baghdad. To the east of center, three platforms are ingressing along a route. When they reach the end of the route, a course is plotted toward a target of interest. Just west of the center of the city, a B-2A bomber is about to release a short range bomb on a biological weapon facility. After releasing the weapon, it turns around and egresses along Path 11. Also, for some reason there is a platform in the middle of the city in a very high threat area. This platform is attacking the biological weapon facility south of the city center. Tracing its behavior through the playback file indicates that its course was plotted in this manner due to its starting position close to Path 11 and the persistent threat that remained in other corridors around the target area. Based on the distance and threat, the agent decided to "take its chances" by plotting a direct course over an unknown threat area rather than routing itself over a longer, known, dangerous route. The platform survives and egresses out via a different route.

As a final example of how the playback files are used to understand the agent's tactical decisions that arise from the simply programmed rules, a playback file from the TCT attack scenario is shown in Figure 149. In this example, area dominance munitions are dispensed from controlling UAVs that search the battlefield for time critical targets that move and hide at pre-determined intervals.

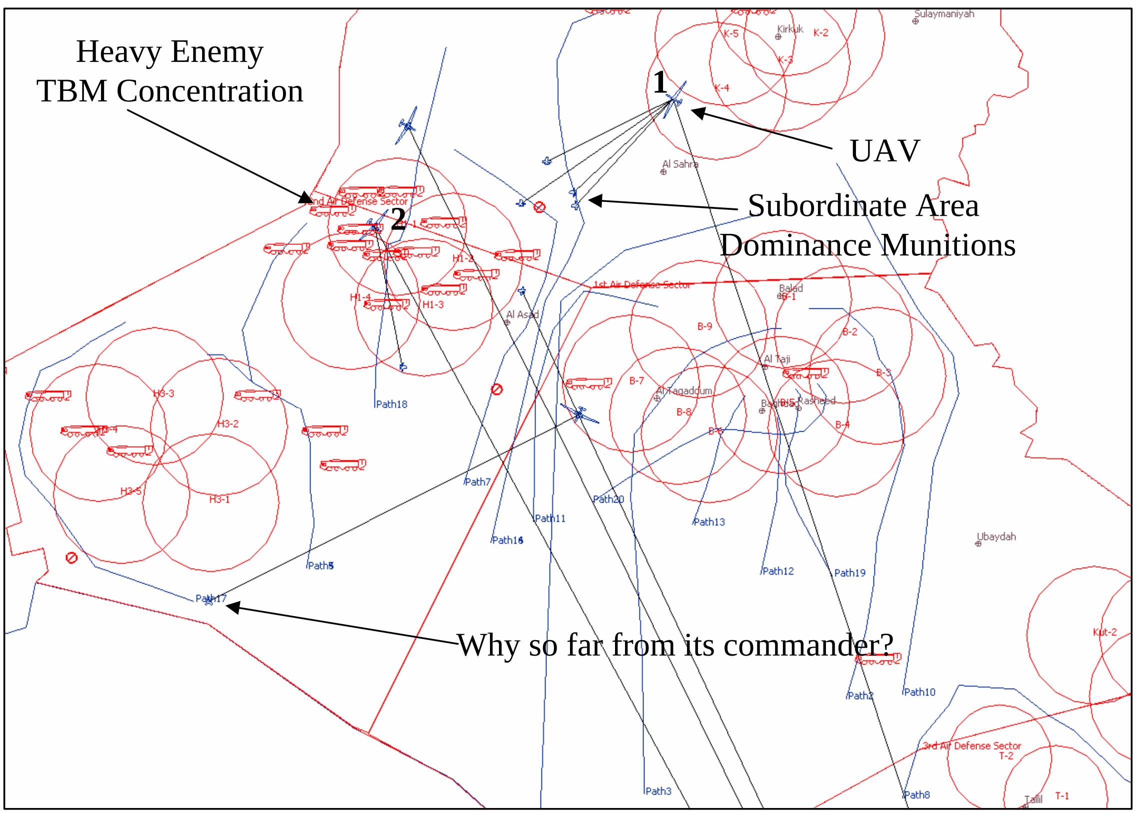


Figure 149: FLASH Scenario Playback File for the TCT Attack Scenario.

The first behavior that is observed is the lone area dominance munition in the southwest corner of the screen. Why is this agent so far from its commander? The munition was released approximately 30 minutes earlier to attack the TBM launcher in the lower left corner. The munition was assigned by its UAV toward the start of the scenario. The observed behavior is the munition flying back toward its commander's search region to dispense its remaining submunitions. It continues to fly until it runs out of fuel.

Also, there is a cluster of munitions released from the UAV labeled "1." These munitions are requested by the UAV labeled "2" which has expended all of its munitions in attempts to kill other targets. UAV "2" is flying a search pattern over a region of high density time critical targets. As it detects these targets, it requests assistance from another nearby UAV that still has munitions. As mentioned in Section B.3, the central tenet of agent-based modeling is the emergence of complex behaviors that result from simple rules and goals programmed into individual agents. The FLASH playback file provides a means for observing the emergent behavior and assessing the validity of the results.

When viewed with subject matter experts and operators, the use of FLASH playback files also stimulates discussion about the scenario and its results. Since many simplifying assumptions are made throughout any analysis process, when experienced analysts view the playback file, their first impulse is to say "that would never happen like that!" This is a good impetus to query the subject matter expert as to how the simulation should be modified to emulate more realistic behavior. Unfortunately, it is very difficult to elicit this information *before* the experts see that the simulation is "wrong." After eliciting this information, the designer should return to Step 3 of the SOCRATES method and ripple the necessary modifications through the analysis process. Iteration through these steps ensures buy-in from subject matter experts and exploits the flexibility of the method.

5.11 Review of the Methodology Demonstration

In this chapter, a ten step methodology for capability-based technology evaluation for systems-of-systems was demonstrated on a testbed problem of interest to the United States Air Force. The first three steps set the objectives and bounds of the study, identified a suitable scenario in the public domain, and developed a series of physics and cognition models to support the simulation of technologies. Step 3 is arguably the hardest of these three steps. While many of the example models included in the FLAMES framework were used as a backbone, much coding and validation of the behaviors in the simulation was required. This step of the SOCRATES methodology is greatly aided by off the shelf simulations and model components that have been created for other initiatives. Integration of many disparate models with variable assumptions and levels of fidelity is sometimes required, but adds to the complexity of the implementation.

In Steps 4-6, a majority of the new methodology development related to the operational level hypotheses is detailed. A method for decomposing strategic objectives to actionable tactical actions was coupled with an intelligent "Meta-General" for battle management. When used in tandem, the targeting and weaponeering functions that are usually driven by human operators is automated to facilitate large-scale technology exploration. This is a critical enabler for quantitative technology assessment of systems-of-systems. Additionally, an innovative method of applying surrogate models to individual agents was implemented. Using these performance-based surrogates, the agent gains a perception of its own abilities in its current environment and alters its decision logic to take advantage of infused technologies.

Steps 7 and 8 set up and execute the necessary code runs to evaluate technologies and create surrogate models around the simulation where appropriate. The use of surrogate models to rapidly perform trade studies on the simulation results is another major advancement in this research. This is the first known application of the technique to a highly complex military system-of-systems simulation. The surrogate model-enabled trade environment highlights the migration from point designs to parametric design space exploration. Step 10 is not "review results" but rather, advocates "what-if?" gaming using the assembled environment. Steps 1-8 are required to generate a single case and get a single result. The additional time required to generate surrogates is minuscule compared to the amount of information that can be obtained from the environment using the simulation-based surrogate models.

This chapter demonstrated that the proposed methodology is a valid means to obtain quantitative results of large-scale military systems-of-systems simulations involving a number of heterogeneous interoperating elements, but only scratches the surface of what can be done with new models, scenarios, and desired capabilities.

CHAPTER VI

SUMMARY AND CONCLUSIONS

"This is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning."

-Winston Churchill

The purpose of this dissertation was to document the development of a methodology for capability-based technology evaluation for systems-of-systems. This topic was driven by an urgent need for a structured method to perform disparate trade studies on potential technologies early in the product life-cycle to facilitate resource allocation toward technologies with the highest payoff towards one or more military capabilities. The research objective was therefore stated as follows:

***Research Objective*: The focus of this research is on the development of a valid, defensible, and practical methodology that facilitates a quantitative assessment of technology potential of systems-of-systems with respect to capability-level gaps and provides information to decision-makers early in the design process.**

First, a methodology was proposed in Chapter IV and developed through a proof-of-concept exercise in Chapter V. Using the testbed modeling and simulation environment created in Chapter V, a series of surrogate models were created that enabled quantitative assessment of potential technologies with respect to top-level capabilities. Since the SOCRATES method fulfils this basic condition, it is a **valid** technique for capability-based technology evaluation.

The traceable, structured analysis enabled by a linked hierarchical series of surrogate models constructed around a simulation framework form the basis for **defensible** engineering conclusions. Any assumption can be identified and changed. Sources of error can be traced both within models and at the surrogate level. In fact, the error metrics for all equations developed are contained in this dissertation. A key attribute of the SOCRATES methodology is that the data speak for themselves. Accurate models, properly constructed surrogates, and effective visualization are used to defend conclusions made by the analyst.

The **practicality** of the method depends on the reason for the study and the resources available. The work performed to validate the SOCRATES method was conducted by a single individual and involved the construction of many experiments, some of which are chronicled here. For this reason, many of the steps in the process were executed in a serial manner; however, steps 1-3 can usually be performed in parallel. The same is true of steps 4-7 and steps 8-10. Simple quick-turnaround studies that do not have a heavy dependence on the scenario may not require the SOCRATES method. Furthermore, in the rare cases where tactics play a negligible role in technology implementation, steps 4-6 may not be needed to perform capability-based analysis. The SOCRATES method is preferred to a brute-force approach to analysis-of-alternatives, but is not a panacea that can be applied to any military SoS technology study.

As a somewhat surprising observation, application of the SOCRATES methodology does not immediately yield clear answers to capability-based technology assessment problems. At the start of this research, such answers were completely obscured by the complexity of the problem. By developing and applying the proposed methodology, some "answers" are now less obscured by this complexity. The structured process advocated in this work provides a means to wade through multiple degrees of freedom and develop multidimensional tradeoff studies that provide more information at the early phases of the design and acquisition process for large-scale systems-of-systems. Further research must delve deeper into the complexity quagmire. A key challenge is the translation of complex information into graphical and visual terms that human analysts can understand and use to make resource allocation decisions.

6.1 Summary of Technology Evaluation Results

Given that the purpose of the SOCRATES methodology is to provide decision makers with insight into technology allocation decisions with respect to capability-level MoEs, it is necessary to review the observations made in the example technology assessment exercise.

The first major observation is that the results are scenario dependent. Changing basing assumptions, enemy threat laydown, force structure, weapon loadouts, and the C2 architecture can dramatically alter the results. This underscores the need to assess potential technologies across a range of scenarios before making resource allocation decisions.

Interestingly, the results fall into two camps: intuitive and non-intuitive. Some of the more intuitive results can be summarized as follows:

- Traditional wisdom says that stealth is an important characteristic of next-generation aircraft; however, this is only true when standoff weapons are not employed. If munitions with sufficient standoff capability can be employed in large numbers and with appropriate precision and lethality, the "missileer" concept (see Section 2.4.3) may be highly desirable. Although individual munitions may be extremely expensive, thousands of munitions may be expended for the purchase price of a single stealth bomber.
- On the other hand, standoff weapons are unlikely to be able to deliver extremely large payloads required for some heavy bunker-busting weapons. The HDBT mission may be a niche capability best served by ballistic missiles with heavy, penetrating warheads that impact at extremely high velocities.
- Persistence in a denied access environment enables the use of lower cost close-range precision munitions and also allows the platform to search for fleeting targets. "Stealth technologies" contribute to survivability in this regime; however, further analysis is needed to determine the best mix of airframe shaping, absorptive coatings, and onboard/off-board jamming.
- High-speed, long-range munitions show an expected benefit in engaging time critical targets. Long-range munitions of all speeds provide a benefit in engaging stationary targets. Further analysis must examine the flight dynamics of these weapons and the

integration of the sensor to provide target tracking during high-speed flight conditions.

While intuitive results tend to confirm that the integrated set of models is doing the right thing, they also do little to help aid the decision maker. Dr. Thomas A. Cruse, Chief Technologist of the Air Force Research Laboratory asked in 2005, "What does this process do for me that I don't already have? How does it reveal something non-intuitive?" [114]. Cruse's comments highlight the importance of being able to use simulation and agent-based modeling techniques to discover non-intuitive results. Some of these interesting conclusions include:

- The importance of the sensor for the time critical strike mission was far greater than anticipated. This implies that the "find" portion of the kill chain is much more difficult than the "engage" portion for these types of missions.
- Increasing platform speed does not provide a universal increase in performance. There is a particular speed where the constraints of drag and fuel consumption intersect to provide the best mix of speed and range. Originally it was anticipated that increasing platform speed would always increase effectiveness. This implies that there is a scenario-dependent balance between platform speed and munition speed/range that must be further quantified with higher fidelity models.
- Platforms and munitions with high success rates required ridiculously low TSFC. The natural progression of propulsion systems to increase flight speed is turbofan, ramjet, scramjet, and rocket. Unfortunately, each change in propulsion system architecture tends to increase the TSFC. Perhaps a dramatic change in propulsion system architecture to focus on low TSFC rather than high thrust is needed to enable high-speed flight. Further exploration of revolutionary concepts such as pulse detonation engines or alternative fuels may be needed to enable next-generation high speed flight.
- Area dominance munitions were expected to be the single greatest "game changing" technology evaluated in the proof-of-concept exercise. Surprisingly, area dominance munitions were outperformed by standard aircraft loitering outside the hostile country. This was traced to the small number of munitions used. To achieve maximum

effectiveness, area dominance munitions must be collaborative, use netted communications, and be employed in large numbers over the area of interest. The cost of such a CONOPS is a factor to consider in their employment.

As the systems-of-systems nature of the problem is modeled in more detail, the number of non-intuitive results is expected to increase dramatically. A quantitative assessment methodology based on modeling and simulation can address issues such as the lack of empirical data for next-generation weapons systems concepts, the uncertainties associated with rapidly adapting enemies, and the complex issues that evolve in the study of network centric operations.

While the aforementioned conclusions highlight the utility of the SOCRATES method, a key advancement is the ability to perform tradeoffs across multiple degrees of freedom, integrate multiple models at various levels of fidelity, and visualize the kinds of tradeoffs that were previously not possible with other technology evaluation methods.

6.2 Subjective Evaluation of the Methodology

Section 1.2 reviewed some existing methods for technology evaluation, and Section 1.2.8 evaluated these methods based on a set of important attributes. The SOCRATES method is compared against the same baselines in Figure 150.

The primary focus of the SOCRATES method is on the quantitative analysis of technologies. The quantification of technology potential and a traceable analysis of technologies to capabilities and vice versa is enabled through the use of a hierarchical object-oriented constructive simulation environment. The use of surrogate models promotes both flexibility and reusability. First, a wide range of studies can be performed by parametrically varying SoS elements and assessing their impact on top level capabilities. Second, the environment is reusable and the generation of multiple surrogates under different conditions results in a library of models that can be used for a variety of studies. Placing the myriad of required assumptions parametrically under control of the designer also promotes reusability as hostile capability adapts and changes over time. These four factors are all rated as "excellent" based on the specific focus of methods development on addressing challenges related to

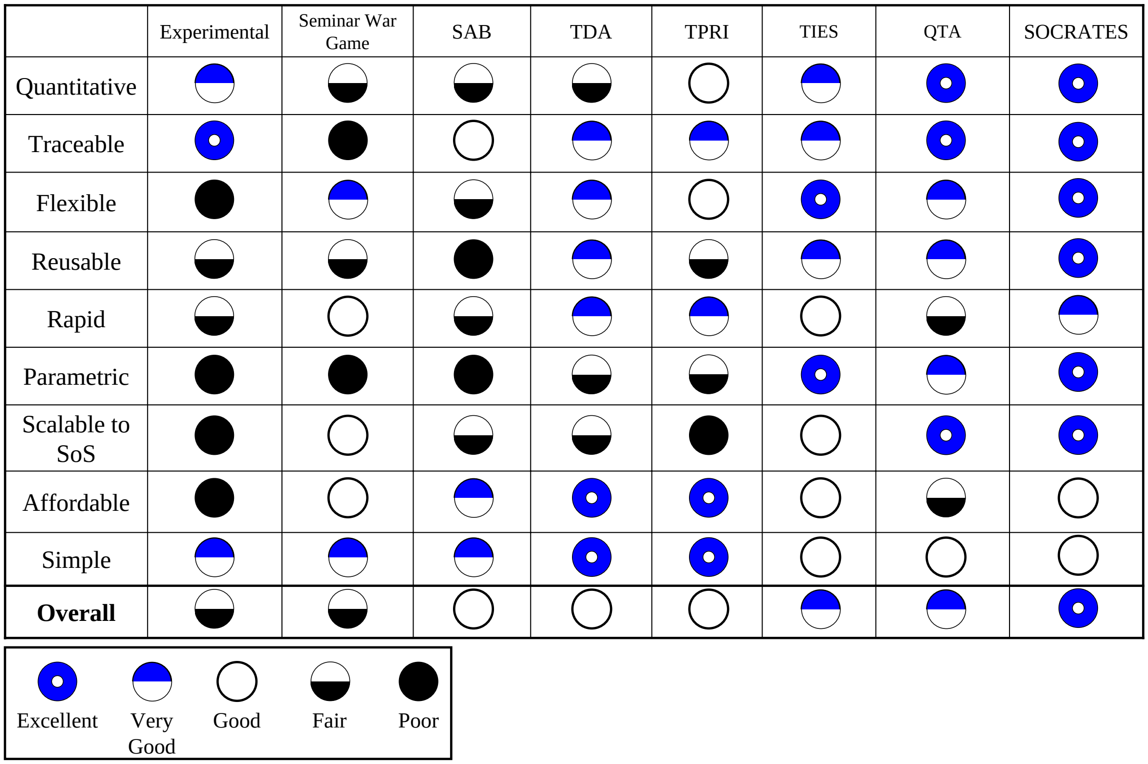


Figure 150: Qualitative Comparison of the SOCRATES Methodology and Other Technology Evaluation Techniques.

these attributes.

Since the method relies on the creation of models and the development of a simulation, even when models are reused from other studies, the speed of the method can never be classified as excellent. The use of surrogate models provides a benefit in this dimension, but the construction of the underlying physics-based analysis environment is a necessary penalty on speed.

The hierarchical surrogate-enabled tradeoff environment developed using the SOCRATES methodology demonstrates both the parametric nature of the method and the scalability to large-scale systems-of-systems. The work in this dissertation combines elements from both TIES and QTA to synthesize a method that contains best-in-class elements of each.

Finally, the proposed methodology can only be rated as “good” in terms of affordability and simplicity due to the complexity of the modeling and simulation environment and knowledge of techniques such as neural networks and advanced visualization. The SOCRATES method is always outdone by qualitative methods in terms of simplicity, speed, and affordability; however, the primary focus of the method is on high quality, traceable, quantitative answers that these other methods lack.

Through the combination of desirable elements from the aforementioned techniques and the infusion of new methods from other disciplines, the SOCRATES methodology emerges as an excellent option for capability-based technology evaluation for systems-of-systems. Over time, further cross-fertilization will undoubtedly lead to the development of new and different methods to address challenges in this class of problems; however, based on the results of the proof-of-concept exercise, the SOCRATES methodology demonstrated applicability to the challenge problem provided.

6.3 When Should SOCRATES Be Used?

Technology evaluation requires time and resources that could otherwise be spent developing the technologies themselves. Building a modeling and simulation environment, executing a design of experiments, and playing "what-if" games certainly takes longer and costs more than ignoring the process altogether. The SOCRATES method is useful when technology solutions themselves are different architectures, which is the case for systems-of-systems in a net-centric context. Whenever the interactions between systems are difficult to quantify using expert opinion or seminar war games, SOCRATES provides a structured process to trace technology impacts to capability level metrics.

According to Brown, technology analysts need to "pick the right tools for the right questions to answer in the right time with the right level of fidelity" [75]. Steps 1-3 of the SOCRATES methodology speak to this issue and steps 4-10 of the process explain how to carry out a technology forecasting activity in the context of a systems-of-systems problem.

6.4 Review of Lessons Learned

The SOCRATES methodology relies on the fusion of techniques outlined in the hypotheses in Chapter III. Through a proof-of-concept exercise in Chapter V, technology evaluation for a Long Range Strike system architecture was demonstrated. The relation between the proposed hypotheses, research questions, and technical challenges are concisely summarized here:

- A "virtual laboratory" using a hierarchical, object-oriented, constructive simulation framework encapsulated within neural network surrogate models can be used to rapidly and quantitatively trace the benefit of proposed technologies to top-level capabilities and vice versa.
- Surrogate modeling approaches can be used not only around the simulation to decrease analysis time, but also inside the simulation to provide intelligence to individual agents and battle-management super agents.
- It is possible to leverage advances in machine learning and artificial intelligence to provide a first-order approximation for human decision making, thus enabling large numbers of simulation runs without manual intervention.
- Intelligent decomposition of problem elements using systems engineering techniques, morphological analysis, SysML, and others leads to analysis simplification without losing the essence of the problem.
- Graphical visualization techniques are needed to understand the volumes of data generated for technology assessment.

The most important lesson learned in the implementation of the methodology is to simplify the analysis framework when possible and avoid unnecessarily complex problems. Even if they are "solved" by the experiment, the analyst never knows it because the complexity of the problem obscures the answer. A "spiral" approach to modeling and simulation where simple models evolve into a higher fidelity analysis with more elements is highly recommended.

6.5 Revisitation of Research Questions and Hypotheses

The SOCRATES methodology resulted from the infusion of techniques and methods proposed as hypotheses. These hypotheses directly address research questions that arose in a review of the technical literature to develop an understanding of the challenges associated with the example problem and the roadblocks to its solution. The answers to the research questions are reviewed here in detail:

1. How can the impact of technologies infused at the system level be analyzed at the system-of-systems level and compared to measurable performance metrics related to capabilities?

An object-oriented, hierarchical, constructive simulation is used to trace system level technology impacts to the system-of-systems level using quantitative MoEs. Physics-based models enable quantitative analysis, k-factors assist in mapping physical parameters to notional technologies, and the inverse design/multivariate analysis technique enables traceability across multiple hierarchical levels. (Hypotheses 2.1, 4.1, 4.2, and 4.3)

2. How can military simulation runs be executed without a human in the loop to make strategic and tactical decisions?

Intelligent agents can simulate basic strategic and tactical decisions both at the battle management level and at the implementation level. A neural network equation enables rapid analysis of potential alternatives and surrogate models at the platform level facilitate technology exploitation. Although it was initially assumed that the intelligent agents would learn and adapt to new situations and develop new tactics on-the-fly, adaptive agents confound technology analysis. A "dumb" agent with a good technology may be outperformed by a "smart" agent with a sub-par technology. This would lead an analysis to wrongly conclude that the inferior technology was in fact worth investing in. An approach where agents were imbued with intelligence *a priori* and then exercised in a simulation addressed this issue.

After the "Meta-General" and exploiting agent concepts were developed, the QFD technique for prioritizing military objectives was developed to simplify the translation of strategic objectives into actionable tasks that agents could perform. The linkage of these three operational level hypotheses addressed the challenges posed by research question 2. (Hypotheses 2.2, 3.1, 3.2, 3.3, and 4.8)

3. Should the goal of the methodology be to identify an optimum technology portfolio that maximizes effectiveness or to seek a balanced portfolio that is robust across envisioned operating conditions?

Although it seems like this research question answers itself, there is a wide variety of literature available on systems-of-systems optimization. This is a combination of operations research techniques and extensions of optimization algorithms developed in the multidisciplinary optimization community to new problem domains. In the case of technology evaluation for evolving military capabilities, neither of these is appropriate or desired. A technology portfolio must be balanced across operating conditions, and theoretically balanced across multiple scenarios and for different capabilities against different threats. This research question is addressed with anecdotal references because the computational resources of the day do not permit full treatment of the aforementioned degrees of freedom. (Hypothesis 2.3)

4. How can the scale of the problem be appropriately reduced without losing the essence of the problem?

Engineers by their nature worship complicated things. Where systems-of-systems are concerned, this approach is a recipe for disaster. The large scale of the problem and the high degree of interactions necessitate an approach to limit the analysis activity to the significant few degrees of freedom. The concept of the matrix of alternatives was used throughout this work to scope the problem, and even to scope the methodology. It is a very simple and convenient technique for enumerating the potential options and highlighting those selected.

When it comes to the development of the conceptual model and the implementation of software code to realize the desired behaviors, some examples from the SysML were used to indicate how the problem can be further scoped. SysML diagramming is becoming more widespread in systems engineering with the recent release of the full standard. A greater use of this technique is recommended when distributed simulation development is necessary. (Hypothesis 4.5)

5. For a given problem, what is the best way to determine the necessary elements of a system architecture?

Here, the need for a solid foundation in systems engineering is demonstrated. Functional decomposition, SWARMing, and brainstorming techniques are all critical functions for problem definition, concept selection, and program management. The word "best" in this research question is slightly disconcerting, because there are many other ways to define a system architecture for technology identification. The easiest way is certainly to receive a list of what is to be modeled in a specification from the customer; however, such detail is seldom available. In this dissertation, functional decomposition is likely the best of the identified techniques at tracing function to form. This helps limit unnecessary complexity in the simulation and is key to facilitating the calculation of capability-based MoEs without the confounding influence of unnecessary simulation elements. (Hypothesis 4.6)

6. How can the simulation process be sped up to allow examination of the design space in a reasonable time frame?

Despite the fact that the number of transistors on an integrated circuit doubles roughly every two years, design studies that took months in the 1960's still take months. Instead of obtaining the same information in less time, analysts leverage advances in computing to obtain more information in the same time. Inversely, the amount of information available in a unit time is proportional to computing resources and how they are used. Surrogate modeling approaches enable high-fidelity modeling at high speeds. The implementation of surrogate models around the simulation environment allows off-design cases within the original design space to be very quickly analyzed, reducing the number of trips back to the computer cluster to re-run cases to observe new behaviors. As another side benefit, surrogate models can also be exercised to visualize the results with relative ease. Surrogate models are the single most important enabling technique in this dissertation. (Hypothesis 4.7)

7. What sampling techniques and modeling techniques are valid for non-linear systems-of-systems simulations?

Designers are always in search of the "perfect" technique to address a certain class of problems. Usually, comparisons of similar techniques in the same field result in only small performance improvements; however, in this research, neural network surrogate models offer marked improvements in accuracy over traditional polynomial response surfaces due to the nonlinear behaviors observed throughout the design space for the military system-of-systems. Their primary drawback is the increased complexity to create and the large number of cases needed. A space-filling design based on a sphere-packing scheme was experimentally found to be the most appropriate for generating surrogate models for this problem domain. (Hypotheses 4.8 and 4.9)

8. How can the importance and sensitivity of individual elements (or degrees of freedom) of the system architecture be evaluated?

The first and most logical choice to address this research question is the implementation of the ANOVA procedure across the hierarchical modeling and simulation environment. Subsequently produced Pareto charts provide a means for understanding the sensitivities of capability-level metrics on meaningful changes made at the system and subsystem level. The only drawback to the ANOVA approach is that the assumptions and the ranges of variability on the input parameters may change, and this can alter the character of the Pareto chart.

The Prediction Profiler was introduced as a dynamic, graphical means for evaluating the sensitivity derivatives across the hierarchical model. In fact, when neural networks are used with the prediction profiler, these sensitivity derivatives may not even be linear or curved: they may take on dramatically different shapes in different areas of the design space.

Unfortunately, the use of this tool for solution discovery can be time consuming because it functions as a one-variable-at-a-time calculator. Essentially, the prediction profiler enables the visualization of the partial derivatives of capability with respect to all contributing metrics while all assumptions and design/technology factors are held constant.

The most powerful (and sometimes most confusing) of the tools used to perform sensitivity analysis is the multivariate analysis tool, used throughout this dissertation as a means to view the total derivative of capability variation as all variables are simultaneously changing. While this view shows an unprecedented amount of information, interpretation is never straightforward. Many factors are changing at once, and the variability must be carefully traced to its causal factors. (Hypothesis 4.10)

9. How can uncertainty be quantified for this class of problems?

While uncertainty quantification is not the primary focus of this work, Section 5.10.6 shows how Monte Carlo simulation can be used with neural network surrogate models to rapidly perform design exploration studies and quantify technology uncertainty both on input variables and output metrics. (Hypothesis 4.11)

The matrix of alternatives from Chapter III is revisited here in Figure 151. As previously mentioned, there are over 4.15 x 10^{22} different methodology options enumerated in this matrix. The selected options identify only a single thread through the matrix and define the SOCRATES methodology outlined in this dissertation. Over time, new alternatives may be developed and cross-fertilized from other fields. If the methodology becomes accepted, the options in the matrix of alternatives will converge to a set of best practices applied to capability-based technology evaluation for systems-of-systems as it is applied to a wider array of problems.

	RQ	Hyp						
Strategic Postulates and Assumptions	1	2.1	Hierarchical System-of-Systems	Yes	No			
	1	2.1	Level of Heterogeneity	None	Low	Medium	High	
	1	2.1	Type of Simulation	Live	Virtual	Constructive	Interactive	
	1	2.1	Programming Approach	Custom (hardcoded)	Object-Oriented			
	2	2.2	Make Decisions in Simulation	Human-in-the-Loop	Decision Tree	Computer Assisted	Artificial Intelligence	Other
	3	2.3	Methodology Focus	Optimization	Robustness	Other		
Operational Hypotheses	2	3.1	Prioritize Targets	Random	Experience	QFD	OEC/MADM	Other
	2	3.2	Incorporate Tactics	Hold Constant	Include All	Use Static Mapping	Optimize for Each Technology	Other
	2	3.2	Type of Battle Manager Agent	Reactive	Deliberative	Mixture	Human	
	2	3.2	Battle Manager Learning Algorithm	Adaptive Neural Network	Genetic Algorithm	Bimodal: Training/Analysis	Random	None
	2	3.3	Type of Subordinate Agent	Reactive	Deliberative	Mixture	Human	
	2	3.3	Asset-Level Decision Rules	Hardcoded If-Then Statements	Random	Performance Vector of Attributes	Response Surface Equations	Other
	2	3.3	Asset-Level Learning Algorithm	Adaptive Neural Network	Genetic Algorithm	Bimodal: Training/Analysis	Random	None
Tactical Assertions	1	4.1	Type of Models	Physics-Based	Empirical	Hybrid	Other	
	1	4.2-3	Analyze Technologies	k-Factors	Unified Tradeoff Environment	Other		
	1	4.4	Trade Study Attributes	Point Design	Bottom-Up	Top-Down	Middle-Out	Other
	4	4.5	Reduce Scale of Problem	Committee Approach	SysML	Matrix of Alternatives	Other	None
	5	4.6	Determine Elements of Architecture	Provided by Customer	Functional Decomposition	SWARMing	Brainstorming Tools	Other
	6	4.7	Speed Up Processes	None	Linear Approximations	Qualitative Mapping	Surrogate Models	Other
	7	4.8	Type of Surrogate Models	Polynomial Response Surface	Neural Networks	Radial Basis Functions	Kriging	Other
	7	4.9	Sample from Design Space	Random	Full Factorial	Box-Behnken	D-Optimal	Uniform
				Orthog. Array	Space-Filling	Central Composite	Latin Hypercube	Other
	8	4.10	Evaluate Importance/Sensitivity	Committee Approach	Prediction Profiler	ANOVA/ MANOVA	Partitioning Techniques	Other
	9	4.11	Account for Uncertainty	Monte Carlo	Quasi-Monte Carlo	Petri Nets	Markov Chains	Other

Figure 151: Matrix of Alternatives for Strategic, Operational, and Tactical Research Questions.

6.6 Recommendations for Future Work

"Let your great object be victory, not lengthy campaigns."

-Sun Tzu

If you ask an engineer how many more cases he needs to run, the answer is always "just one more." On the other hand, if you ask an architect how many bricks he needs, the answer is "one less than too many and one more than not enough." While this dissertation could in theory go on forever, there will always be another building, another mountain, and another challenge for another day. Some suggestions for future work are offered here.

The development of a methodology for capability-based technology evaluation for systems-of-systems was confounded by the large number of degrees of freedom. Most of these dimensions had to be fixed to produce meaningful results with the resources available. The first potential avenue to explore is a comparison of multi-domain solutions such as air vs. land vs. space. These were not examined in this work due to the disparate models needed to compare multi-domain concepts. Another area of interest is the expansion of the agent-based tactical exploration. Can agents develop new tactics? Can they share the tactics with other agents? Multi-agent learning and evolution is a fast-growing area of research. Political dimensions such as overflight rules, basing restrictions, and rules of engagement would provide an interesting diversion in international affairs and technology policy. The assessment of a single capability across multiple geographic regions with varying terrain is a logical extension of the demonstration in this work. The same can be said of developing an integrated suite of systems that provide multiple capabilities. This is, after all, more like the idea of designing "a military" instead of one element within it. Other treatments could include a capability analysis of legacy systems with new technologies. Decisions on when to decommission existing systems and spiral in new systems and capabilities is a good collaborative area for operations researchers, aerospace engineers, and technology developers.

On the hard-core physics-based modeling side, it would be of interest to see how the aforementioned methodology can be combined with high-fidelity models such as CFD aerodynamics codes for hypersonic drag calculations, finite element models for missile and aircraft structures, or six degree-of-freedom trajectory tools for missile and aircraft flight mechanics. Control systems received no treatment in this work, and propulsion systems got less attention than originally planned. The rigors of developing the methodology for a simple test case prevented resources from being allocated to these issues, but that does not mean they are not of interest.

The methodology demonstrated dealt largely with the evaluation of effectiveness. Little consideration was given to affordability and life-cycle cost models. Future work should include MoEs related to these factors and attempt to formulate a holistic life-cycle cost model that addresses investment concerns, budgetary concerns, and operational issues. Furthermore, the analysis performed in this work assumed that all operational assets had the equipment, munitions, and fuel they needed to prosecute their mission. Adding the umbrella of logistics support brings a complicating but critical element into the fray. The design of a system-of-systems robust to resupply shortages, denied basing, changing alliances, and fuel restrictions would be of great interest to the operational community.

In terms of methods development, a greater treatment of uncertainty and its quantification is desired. An easy-to-use yet intuitive means to quantify uncertainty at multiple hierarchical levels would add value to the existing process. Other surrogate modeling techniques such as Gaussian Processes may offer advantages in some problem domains. Agent-based modeling was chosen over the complimentary system dynamics approach for battle management; however, can a more generic framework be constructed with the latter?

This section not only highlights the myriad of opportunities in the study of systems-of-systems, but also underscores an important life lesson: no matter how hard you work and how much you do, you are only scratching the surface of scientific discovery.

6.7 *Concluding Remarks*

"I am the wisest man alive, for I know one thing, and that is that I know nothing."

-Socrates

The quote by Socrates is apt to describe the dilemma faced by systems-of-systems analysts: problems facing the community seem intractable due to the complexity of the problem. This complexity is inherent in our daily lives from weather to traffic to the movement of financial markets. A scientific understanding of the underlying phenomenology of the interconnected web of systems all around us is a long way off.

Just because something is difficult does not mean it is unworthy of study. In this dissertation, methods and techniques to account for complexity in large scale systems-of-systems were explored. A methodology to enable quantitative technology evaluation of systems-of-systems using modeling and simulation was successfully demonstrated for a Long Range Strike system architecture. The dynamic tradeoff environment created in this demonstration is useful for decision making and analysis of large volumes of data in an interactive visual manner.

But what about the answer? What *is* the best mix of technologies for a future Long Range Strike System? Unfortunately, direct answers are hard to come by; however, the proposed method provides insight into the problem that was previously obscured by layers of complexity.

For example, one discovery is that the analysis of technology sensitivities is inherently more useful than direct technology impacts. Since so many degrees of freedom are available, locking them down for the purpose of getting an answer has almost no meaning. In fact, there is literally no analyst or decision maker at any level that can specify the values of all the unknown parameters simultaneously. The SOCRATES method facilitates the creation of a parametric tradeoff environment to direct discussions about the sensitivities in different regions and facilitate "what-if" games between technologists and decision makers.

The power of visualization cannot be underscored. The result of an analysis is just a table of numbers. Trends are invisible until they are presented in a visual manner. Using dynamic interfaces, some of the assumptions can be analyzed parametrically and the results can be displayed visually. The inherent processing power of the human brain is far more useful than any number of Pentiums.

Finally, the use of a computer-simulated alternate reality for generating data and drawing conclusions in a "virtual laboratory" is a capability that analysts only ten years prior could only have dreamed of. As the techniques developed in this dissertation are combined with advances in computing power, modeling software, and the cognitive sciences, one can only wonder how the problems of the future may be addressed.

APPENDIX A

REVIEW OF STRATEGIC HYPOTHESES

A.1 Hypothesis 2.1: Quantitative Technology Evaluation Using a Hierarchical Object Oriented Constructive Simulation

Although the overall desire of the proposed methodology is to quantitatively assess the impact of technologies at the system-of-systems level, can this even be done analytically? Several notable questions arise:

- Is the system under test a hierarchical system-of-systems?
- How heterogeneous are the elements of the system-of-systems?
- What type of simulation should be used?
- What programming approach is desired?

A review of existing methods for technology evaluation for systems-of-systems identified a dearth of techniques that quantitatively assess the impact of technologies across hierarchical levels. Systems-of-systems are dominated by interactions between components and emergent behavior that is difficult to predict with qualitative methods. As a result, the need for quantitative evaluation of technology impacts against top-level capabilities necessitates a modeling and simulation environment. A literature search was conducted to understand the type of simulation needed, decode the detailed jargon of the field, and identify simulation tools that facilitate analysis goals.

A.1.1 Defining the Lingo: Introduction to Military Simulation Terminology

In the military simulation community, it is important to make clear the distinctions between terms, as each has a specific meaning attached to it. The purpose of this section is to delineate between terms that are used throughout this work and establish a common basis of understanding through definition.

A.1.2 Understanding Effects, Capabilities, Tactics, and Strategy

> *"All men can see the tactics whereby I conquer, but what none can see is the strategy out of which victory is evolved."*
>
> -Sun Tzu

The military modeling community is rife with lingo, acronyms, and terminology that must be crisply defined to communicate effectively across companies and government institutions. The first step in formulating a methodology is developing the linguistic database to understand the issues associated with military modeling. For example the terms *policy*, *doctrine*, and *strategy* are often confused and sometimes used interchangeably. While these terms are all distinct, they are interrelated.

- **Policy** is guidance that states what is to be accomplished [431]. Policy is dictated at the national and international level by the President of the United States (POTUS), Congress, the United Nations, or other international bodies such as the International Atomic Energy Agency (IAEA). National security policy is documented by the POTUS in the National Security Strategy (NSS) while military policy is defined by the Secretary of Defense in the National Defense Strategy of the United States of America and the Chairman of the Joint Chiefs of Staff in the National Military Strategy of the United States of America [80, 360, 306][1]. Furthermore, "within military operations, policy may be expressed not only in terms of objectives, but also in rules of engagement" [431]
- **Doctrine** "consists of the fundamental principles by which military forces guide their actions in support of national objectives" [468]. "Doctrine states that airmen should, for example, seek to achieve air superiority, but doctrine does not focus on what platforms should be used to achieve that effect" [431]. It explains how a job should be performed to achieve an effect.
- The **Concept of Operations (CONOPS)** is "a verbal or graphic statement, in broad outline, of a commander's assumptions or intent in regard to an operation or

[1] Although these documents are identified as "strategy," they are actually statements of policy.

series of operations. The concept of operations frequently is embodied in campaign plans and operation plans" [468]. According to INCOSE, the CONOPS "describes the way the system works from the operator's perspective" [217].

- In contrast to CONOPS which are broad statements that are seldom system specific, an **Operational Concept** is "an abstract model of the operations of a specific system or group of systems, usually developed as part of the acquisition process and used throughout the design, development, test and evaluation (DDT&E) phases of the system life cycle" [42]. A single CONOPS can encompass multiple operational concepts.
- **Strategy**, which comes from the Greek *stratos* (army) and *ago* (to lead), is "the art and science of developing and employing instruments of national power in a synchronized and integrated fashion to achieve theater, national, and/or multinational objectives" [468]. It defines a plan of action, matching means to ends. According to Clausewitz, strategy is "the theory and the use of combats for the object of the war" [100]. Military theoreticians such as Bernard Brodie claim that strategy has been constant throughout the history of warfare [74].
- **Tactics**, from the Greek *tasso* (to arrange), encompass the methods or procedures used to achieve a goal. According to Clausewitz, a tactic is "the theory of the use of military forces in combat" [100]. Tactics can be grouped under strategy. A series of planned tactics employed to reach a desired end state are one method use to describe the concept of operations. Because the adversary makes decisions after each friendly play, the conditions of the game are constantly changing. As such, the exact sequence of tactics planned in the CONOPS may not yield the desired end state. Also, the enemy can quickly adapt if the same tactics are repeatedly used. Unlike strategy, tactics are ephemeral and are often tied to the technology of the day.
- **Effects** are the "full range of outcomes, events, or consequences that result from a specific action" [431]. Direct effects refer to measurable quantities of military progress such as damage caused, ships sunk, etc. Indirect effects refer to quantities that are more nebulous but contribute significantly to warfighting objectives such as morale

and momentum.

- The term **Use Case** has become increasingly popular in industry. A use case defines how a user uses a system. If the user is a pilot, the use cases are tactics or procedures. If the user is a general, the use cases are strategies, operational concepts, or CONOPS.

The system-of-systems triangle shown in Figure 10 can be extended to a pyramid called the "PASS Pyramid." PASS is an acronym that defines the four sides of this pyramid: principles, actions, stakeholders and systems. The relationships between the four sides are illustrated in Figure 152.

Finally, these terms can be further clarified by an analogy to the game of chess. Chess is a game with defined rules, which are a statement of *policy*. This policy defines the objective: win the game through the elimination of the opponent's king. This is the desired *effect*. *Doctrine* dictates that the player uses his pieces to achieve this effect. It does not specify which pieces to use and in which order to use them. *Strategy* will dictate the general disposition of the player, whether he is primarily defensive or offensive or under which conditions he changes disposition. The allowable moves of each piece (or asset) are the *tactics* that may be employed. A knight may employ the tactics of "move two spaces in a direction and one space in a perpendicular direction" whereas bishops may employ only diagonal tactics. Pieces that have more flexibility in their tactical options (for a given state of the game) are generally more effective. A planned series of tactics that proceeds to an end state embodies the *concept of operations* (CONOPS). Since it is not possible to forecast what the opponent will do, the CONOPS may have to be altered depending on the state of the game.

All assets can provide a single *capability*: to eliminate hostile pieces from the board[2]. The degree to which the pieces can eliminate other pieces can be seen as their measure of effectiveness in providing that capability. One would generalize that the queen is therefore the most effective piece on the board, and the pawn is the least effective; however, this is not the case. As mentioned in Reference [454], a capability is provided "under specified

[2]though it could be argued that all pieces except the king can provide the capability "sacrifice yourself to save the king" at various times in the game.

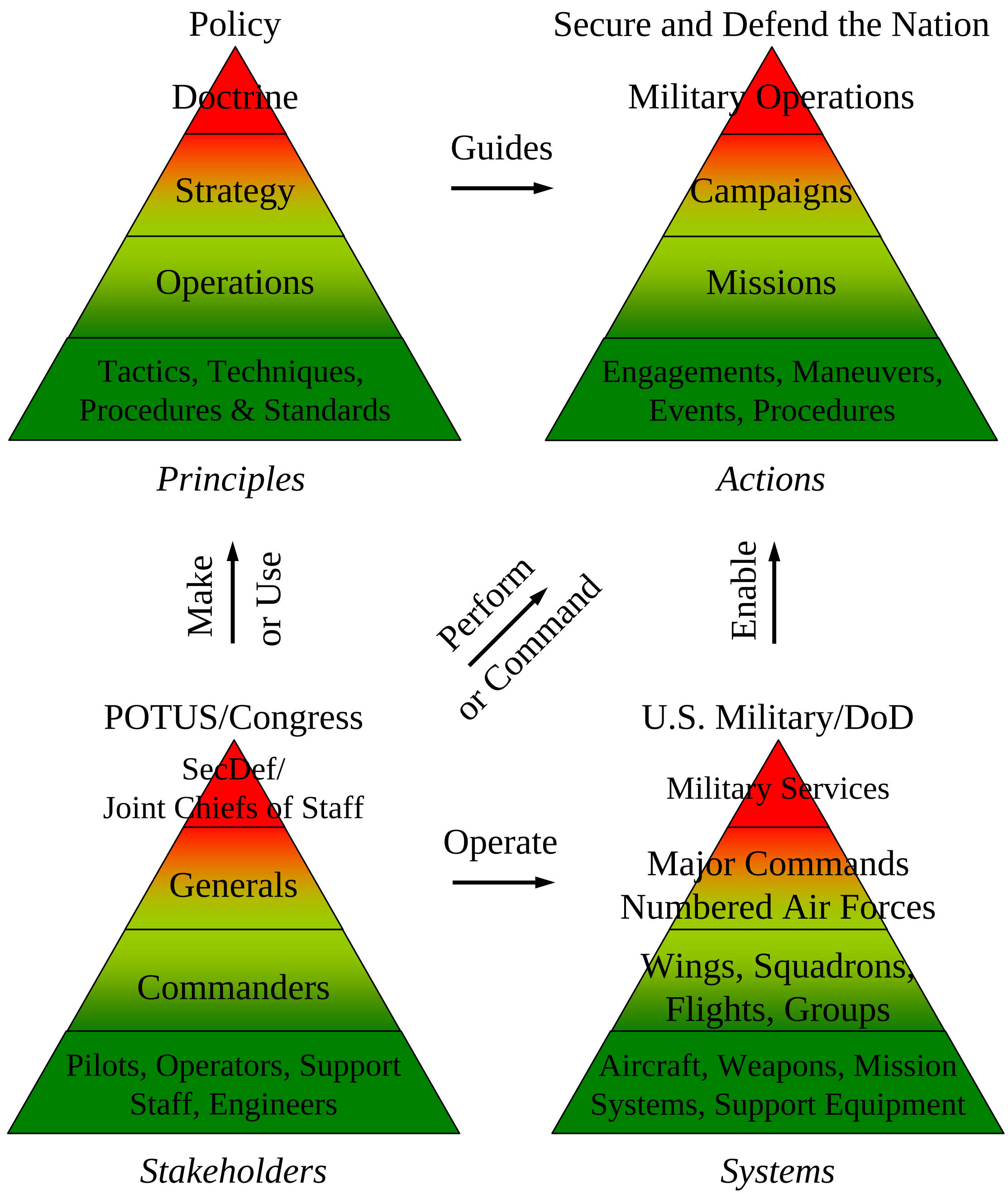


Figure 152: The Principles, Actions, Stakeholders, and Systems (PASS) Pyramid.

conditions." In the opening move of the game the queen cannot move. Therefore, her effectiveness under these conditions is zero because she cannot provide the elimination capability. A queen also has an effectiveness of zero after she is eliminated or when it is the other player's turn.

Alterations of existing pieces to provide new movement directions would be an example of the development of new tactics. One side may employ new tactics with the same assets which could provide an asymmetric advantage depending on the state of the game. Teleportation from one square to any another would be an example of a tactic that would greatly increase the survivability of the king. Although it is difficult to distinguish between technologies and tactics in the game of chess since the pieces are all similar except for their tactical employment, one example would be an increase in speed that allows the friendly player to make two moves in the time the hostile player can make a single move. This "technology" would greatly increase the effectiveness of the faster piece and may outweigh a superior tactical advantage of certain standard pieces. A final example of technology would be a "stealth" ability, whereas the opposing player did not know the location of the king. In this situation, the attacker would have a difficult time developing a concept of operations that centered on the king. A reasonable strategy would be to eliminate all other pieces and then begin to search the board for the hostile king. Clearly, this technology yields a tremendous tactical advantage as it modifies the strategy, CONOPS, and tactics that the friendly player must employ. This situation is similar to the challenge faced in the elimination of time-critical targets in an anti-access environment [68].

A.1.3 Scope and Timescales for Military Simulations

Different types of simulations exist and are often defined by the timescale over which critical events occur.

A **campaign** is "a series of related military operations aimed at accomplishing a strategic or operational objective within a given time and space." A campaign has a geographic location, or **theater**, associated with it. Furthermore, a campaign is considered concluded once an *overall* desired effect is achieved (eg: sufficiently degraded the enemy's offensive capabilities to the point which they are no longer a threat). A campaign is oriented on the enemy's centers of gravity and employs all available sea, air, land, space, and special operations forces in a simultaneous and synchronized manner [372].

The related military operations within a campaign are **missions**: "the tasks, together with the purpose, that clearly indicate the action to be taken and the reason therefore" [468]. The purpose of a mission is also to achieve a desired effect; however a mission generally involves a smaller force structure or a more limited timeframe.

In the course of performing a mission, tactical conflicts may occur between friendly and hostile forces. These are called **engagements**. In summary, a campaign occurs one or more theaters and is comprised of a number of missions which themselves may contain multiple engagements. A one-on-one or a few-on-few scenario can be considered an engagement. When more assets are involved or when there is a clear motivation behind coordinated action, this is usually referred to as a mission. The largest scale operations are considered campaigns.

All three of these actions end when some desired effect is achieved with the strategic importance of that effect increasing as one moves up the hierarchy. The time frame and scope (see Table 32) of the conflict may also indicate which action is taking place:

Table 32: Scale and Time Frame of Three Major Classes of Simulations.

Category	**Scale**	**Time Frame**
Engagement	One-on-One or Few-on-Few	Minutes to Hours
Mission	Several-on-Several	Hours to Days
Campaign	Many-on-Many	Days to Years

A.1.4 Types of Simulations

Simulation can be defined as "the process of imitating a real phenomenon with a set of mathematical formulas" [9]. Simulations allow designers to examine changes on a system without affecting the actual system. Modeling physical systems in a virtual environment provides insight into the behavior of a complex system and its interaction with other systems [126].

Simulations are classified into three distinct types. *Live simulations* involve real people using real (or simulated) hardware in the real world. For example, mock trials may test legal prosecution and defense teams while army units may practice combat maneuvers in the desert with live rounds. *Virtual simulations* are a broad class of simulations that involve real people using simulated systems. The most definitive example of a virtual simulation is a flight simulator. Commercial flight simulation tools approach real-world accuracy. Using virtual simulation, humans can practice unusual or dangerous maneuvers such as Space Shuttle reentry techniques without actually endangering human life. Virtual simulation is widely used to provide realistic and cost effective training aids. The third type of classification used is *constructive simulation*. These simulations rely on simulated people interacting in a simulated world, and while humans often stimulate events in the environment, the outcome is not primarily determined by them [451]. Constructive simulations typically rely on computers to simulate realistic human behavior in a representative environment. Military wargames typify constructive simulation. Many popular computer games including the Electronic Arts™ series of games (*SimCity™, The Sims™, etc*) and *Command and Conquer™* are examples of commercially available constructive simulations.

Interactive simulation, or "human-in-the-loop" simulation, is a type of simulation that crosses the boundary between virtual and constructive simulations. A fully virtual simulation is categorized by *continuous* human input while a completely constructive simulation requires no human intervention once the simulation begins. In a fully constructive simulation, a *SimCity™* gamer could open a saved game file and let the game run without intervention. Similarly, a fully virtual simulation is one in which a pilot would fly in a virtual environment independent of other traffic or air traffic control. Interactive simulations combine elements of both types, for example, to allow human pilots to attack ground targets whose movements are governed by artificial intelligence. These types of simulations are summarized in Figure 153.

It is interesting to note that the final permutation between real/simulated worlds and people involves simulated people operating real-world systems. While not classified as a type of simulation, this model of operation would refer to autonomous systems such as ISR UAVs or intelligent agents such as the Microsoft® Office Assistant interfacing with real

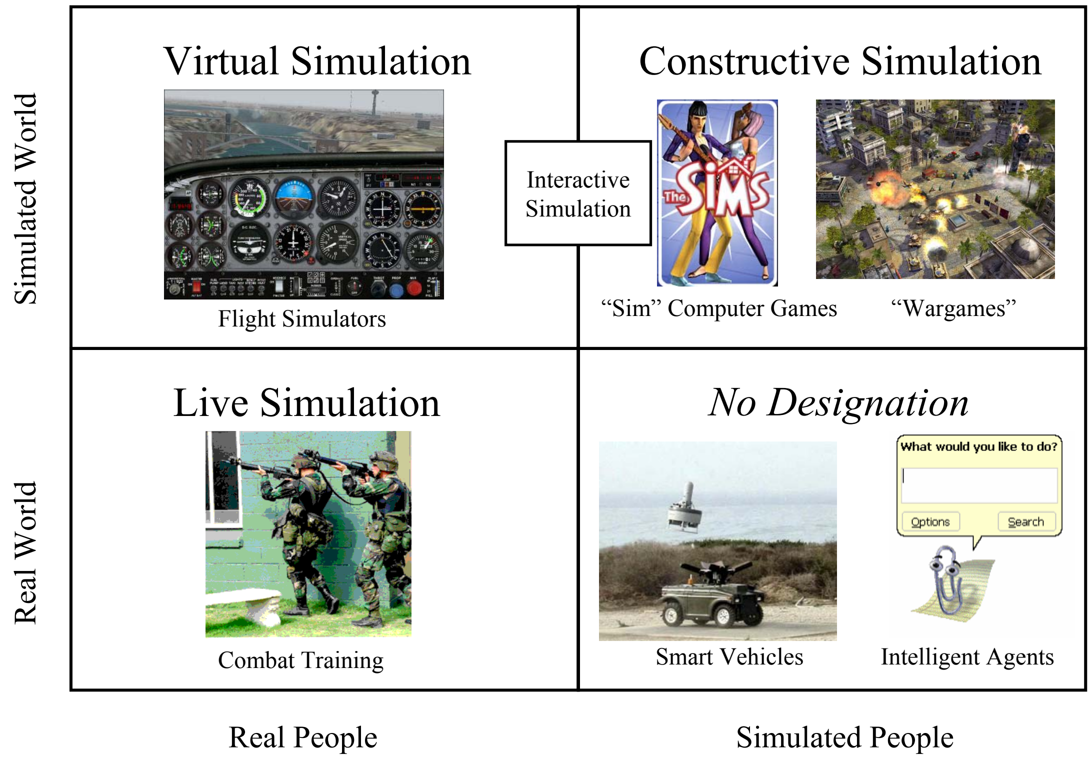


Figure 153: Taxonomy for the Classification of Simulations.

people. It is important to note that it is often difficult to directly classify a simulation as live, virtual, or constructive. Realistic simulations may incorporate elements of different classes. The use of simulation tools for concept evaluation and selection is a continuum of the three major classifications. Different phases of the design cycle rely on different types of simulation, as shown in Figure 154.

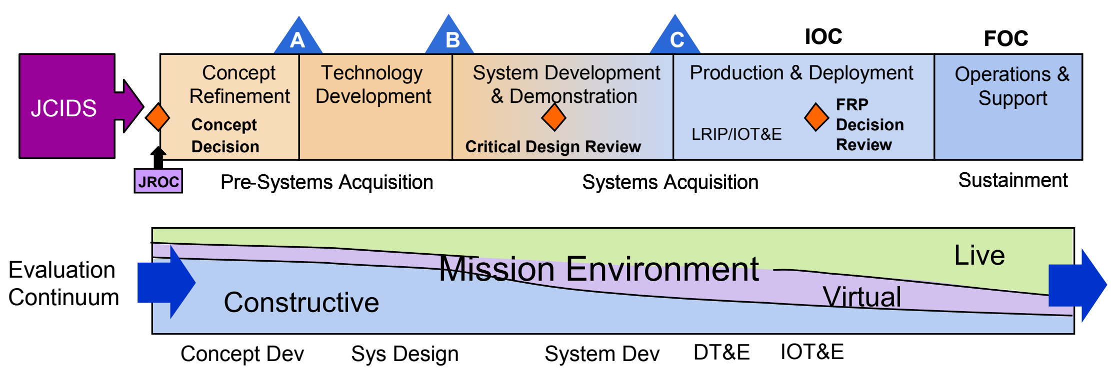


Figure 154: Evaluation Continuum Over the Project Life Cycle [336].

Recently, advanced technology in the form of new munitions has permeated the existing military architecture as a cost-effective means of implementing additional capabilities using legacy systems. Unfortunately, this integration is primarily confined to the system acquisition and sustainment phases of major weapon systems programs. According to Schrage, an IPPD process that provides more detailed information in the early design phases reduces the total cost of ownership while enabling transitional capabilities [369]. As previously mentioned, this process is enabled through computer simulation in the concept development and system design phases. As shown in Figure 154, in this stage of the product life cycle a majority of the simulation activities should involve constructive simulation.

A.1.5 Utilization of Constructive Military Simulations for Technology Evaluation

Recently, research into technology evaluation for systems-of-systems has involved military campaign modeling. Soban used the Integrated Theater Engagement Model (ITEM) to develop a probabilistic system effectiveness framework for aircraft survivability [378]. ITEM is used for joint simulation of air, land, and naval warfare systems has a strong emphasis on visualization and uses a graphical user interface [38]. The key observation made by Soban is that a large amount of computational analysis effort is traditionally spent optimizing an aircraft system with little regard for how it performs in its operational environment. Soban extended design methods for aircraft design such as probabilistics and robust design to the campaign level and demonstrated that various difficulties with model integration and abstraction could be solved through traditional systems engineering approaches applied at the campaign level.

Soban noted that surrogate modeling approaches of the time had difficulty modeling discontinuous behaviors, such when the detection range of a SAM is lowered to the point that the aggressor aircraft are not detected at all. She also noted that code selection impacts process implementation to a great degree. While the PoSSEM methodology developed by Soban is independent of a simulation code, the complicating nuances of ITEM impacted her ability to fully explore desired degrees of freedom. Soban recommends the selection of simulation tools that are flexible, lack a steep learning curve, and can be reasonably integrated into computational frameworks. To make decisions in the PoSSEM proof-of-concept demonstration, Soban used decision trees with finite probabilities resulting in four analysis pathways that required surrogate model generation but notes that for larger problems the number of pathways to explore grows exponentially. Depending on the simulation tool, this technique may or may not be valid.

Frits formulated a system-of-systems-based robust design environment for undersea weapons and developed the necessary tools to demonstrate the impact of tactics on torpedo design [159]. Frits noted that although more than 15 government undersea warfare simulation tools exist, all accurate undersea engagement simulations were classified or unavailable for academic research. To demonstrate the impact of tactics on design, Frits implemented a minehunting simulation in MATLAB® that utilized a time-marching analysis of a notional mine detection, classification, and neutralization vehicle. Frits concludes that research into complex systems-of-systems would be aided by generic engagement modeling tools. To fully integrate tactics into the conceptual design environment, Frits notes that parametric tools that "account for the myriad of tactical decisions possible without resorting to a man-in-the-loop style analysis" are needed [159].

A.1.6 Review of Existing Military Simulations

The Defense Modeling and Simulation Organization (DMSO) maintains a database of all approved military simulations, models, data sources, and other utilities for the analysis and evaluation of military simulations called the Modeling and Simulation Resource Repository (MSRR). Currently, there are over 500 simulation tools approved by the Office of the Secretary of Defense, Missile Defense Agency, Navy, Army, Marine Corps, and Air Force [448]. Of these, thirty-seven simulations are designated by Air Force organizations [425]. The characteristics of major simulation codes utilized by the Air Force are summarized in Figure 155. Of the constructive simulations listed in Figure 155, only FLAMES (highlighted in green) is designed to function from the subsystem to campaign level.

Furthermore, while the identified tools have been approved by the DMSO, a simulation

Characteristics			Type of Simulation			Level of Detail			
Simulation Name	**Sponsor**	**Primary Purpose**	**Live**	**Virtual**	**Constructive**	**Subsystems**	**System**	**Mission**	**Campaign**
AWSIM	AFAMS	Air Theater Model			●				●
BRAWLER	AFSAA	Air-to-Air Analysis			●			●	
CFAM	AFSAA	Weapons/Force Structure			●				●
DIADS	AFMC-AFFTC	IADS			●			●	
EAAGLES	ASC	Human-in-the-Loop		●			●	●	●
EADSIM/Brawler	AFMC	Air Combat MCS			●			●	●
FLAMES	**ACC**	**Any**			●	●	●	●	●
GATER II	ACC	IADS			●			●	
ICET	AFMC-AFRL	Munitions/Techs			●		●	●	
ITEM	CNO (Navy)	Joint Campaign Sim			●			●	●
JCAS	AFAMS	C4ISR/Strategic Attack			●			●	●
JECEWSI	AFAMS	Electronic Warfare			●				●
JIMM	AFAMS	Mission DES	●	●	●			●	
JSAF	AFAMS	RTS			●			●	●
LOGSIM	USAF HQ	Logistics/Maintenance			●		●		
MIL AASPEM II	AFMC-ASC	Air/Air Engagement		●	●			●	
SEAS		C4ISR Modeling			●		●	●	●
SIMFORCE	AFMC-AFRL	C&C		●				●	●
STK	AFAMS	C4ISR			●		●		
STORM	AFSAA	THUNDER Replacement			●			●	●
SUPPRESSOR	AFMC-ASC	IADS			●			●	
TESS	AFAMS	Engagement Sim	●	●	●	●	●	●	
THUNDER	AFSAA	Joint Campaign Sim			●			●	●

Figure 155: Comparison of Major Air Force Simulations (Compiled from data in Reference [425]).

framework must be available for public use and affordable to acquire and maintain. These constraints eliminate several simulation tools from consideration as a framework for the proof-of-concept in this dissertation.

Also, implementing a realistic modeling and simulation environment that allows flow-up of technologies from the engineering subsystem level to capabilities at the campaign level traditionally requires the linkage of a variety of simulations. One criterion that is therefore useful in the selection of a simulation framework that supports integration of models across the simulation hierarchy.

Another criterion in the selection of a simulation framework is the suitability of the tool to facilitate large-scale system-of-systems studies. According to Painter, object-oriented simulations, now the standard in the military simulation community, are appropriate for this task [333].

A.1.6.1 Benefits to an Object-Oriented Approach

Flexible military simulations can be enabled through the use of object-oriented constructs. Object-oriented programming (OOP) is a programming paradigm widely popular in computer science and "brings a discipline to the process of writing code which can produce software that is modular, maintainable, and extendible" [398]. OOP is the computational equivalent to a system-of-systems: the computer program is composed of a collection of individual units called objects. This flexible approach allows objects to send and receive messages in addition to processing data [16]. This is in contrast to the traditional view of programming in which sequential instructions were given to the computer, a result of the punch-card era. It is believed that the term "object-oriented programming" developed literally from the grammatical meaning of the word "object" which is always attached to a verb. Since "object does verb", it is easier to develop simple functions and link them together. Subject-based software has ambiguous and complex requirements that lead to monolithic computer programs. This is analogous to defining a system using functional decomposition (top-down, verb/function oriented) or physical recomposition (bottom-up, noun/subject oriented) [333]. Although there are several different types of OOP, there are some general characteristics that all languages and frameworks share:

1. **Objects**: Self-contained modules that correspond to various aspects of the problem. They include "a local state and the set of operations that are allowed to change that state" [398].

2. **Abstraction**: The ability of objects to focus on essential details only and ignore some aspects of the information that it is processing. Like individuals, the objects can perform a variety of functions depending on their state and the input conditions. They may not need all types of information to achieve a desired function in certain situations.

3. **Encapsulation**: Users can only interact with objects through intended methods. An interface is provided to the user to identify what the input and output methods are. It is not necessary for other objects to see the methods and processes inside the code. This is similar to the way extensive details are enclosed within a response surface model.

4. **Inheritance**: Objects can be defined which are modifications or subclasses of existing objects. For example, a *fighter aircraft* inherits methods and processes from the *aircraft* object, which in turn inherits aspects from the *vehicle* object.

5. **Polymorphism**: Objects can inherit methods from other objects, but their internal routines may cause them to handle those methods differently. For example, if a human receives a command to "move quickly," it moves its legs and begins to run. If a car receives the same message, it throttles up the engine and spins its wheels faster. In this way, desired effects can be easily enumerated in terms of the actions required to achieve them.

The above characteristics provide several benefits which are key to this research. First, the definition of small modules is ideal for various levels of systems required in a system-of-systems approach. Modularity also allows extremely rapid creation of multiple aircraft of different types that may even have slight variations to reflect pilot skill. Second, abstraction plays a key role in the design for systems that perform multiple functions. Nearly all items in a military system-of-systems effectiveness framework perform multiple actions under a variety of conditions. Message passing between modules can simulate network centric warfare or allow objects to operate collaboratively. Third, encapsulation provides a way to simulate the chain of command and account for operational uncertainty. Soldiers are trained according to some known procedures and have certain tactics available. The general tells his troops to perform a given mission (message call to the objects) and they execute the necessary methods with the information they have. The general does not tell each soldier when to arm his weapon, where to point it, and how often to shoot. Next, the idea of inheritance is helpful to rapidly create a computer model of an architecture, as many items have similar methods such as move, fight, defend, and flee. Polymorphism allows easy switching of physical systems that are intended to provide the same effect. Polymorphism and inheritance both help standardize interfaces between code modules.

This section is only a cursory overview of several aspects of this immense field for which countless volumes and entire disciplines have been developed. It is evident that a modeling and simulation environment that utilizes an object-oriented approach is key to successfully developing a large-scale simulation capability for multiple types of interacting systems that must communicate with other heterogeneous assets to achieve a desired effect. OOP also makes the implementation of an agent-based artificial intelligence framework easier (Section A.2.7) and will allow the implementation a variable fidelity approach to subsystem modeling with minimal code updates (Section 5.3.1.7). For all of these reasons, simulation frameworks that use OOP must be utilized or written to support system-of-systems design.

A.1.7 Selection of a Simulation Framework

The implementation of a system-of-systems effectiveness evaluation environment for the conceptual design of large scale military system architectures requires a constructive simulation (see Section A.1.4). Ideally, this tool should be commercially available, object-oriented, valid across the entire military simulation hierarchy, and flexible enough to allow the modeling of any type of system with variable levels of fidelity. After reviewing the commercially available and government-owned simulations shown in Figure 155, only the FLAMES framework by Ternion, Inc. meets all of the standards of effectiveness and availability listed above.

The FLexible Analysis Modeling and Exercise System (FLAMES) is a *framework* used to build constructive simulations that relies heavily on an object-oriented programming approach[3] [397, 28]. By creating objects within the FLAMES framework and defining the linkages between objects, a user can create a constructive simulation for systems and engineering analysis, testing, training, mission planning, entertainment, and a wide variety of other applications. Due to the object-oriented nature of the framework, FLAMES objects (or models) can be created from the engineering subsystem level to the campaign level

[3]Programming purists note that FLAMES is *not* a constructive simulation. It is a *framework* that can be used to build them.

provided that the appropriate interfaces between the various hierarchical levels are defined. FLAMES objects can also be created which define linkages to live and virtual simulations. The framework comes with a number of “example models,” which, when modified and extended can provide reasonable approximations of the systems necessary to demonstrate military architectures. FLAMES has been applied to a wide variety of problems including network centric systems analysis [495], conceptual weapons system design [486], directed energy weapons performance [413], and military engagements [344]. The open source architecture of FLAMES provides the flexibility to model nearly any system at any level of detail, a majority of the industry and government activities using the software are either classified or proprietary at this point.

FLAMES is actually a family of several customizable products based on the C/C^{++} programming language. The products comprising FLAMES include:

- ***FORGE:*** *FLAMES Operational Requirements Graphical Editor*. Graphical interface to the FLAMES database. Allows creation of scenarios, units, and hardware models. The 2-D and 3-D interface lets a user view the scenario as it is created.

- ***FIRE:*** *FLAMES Interactive Runtime Executable*. Executes a scenario created in FORGE. Can run in batch mode and be executed from the command line.

- ***FLASH:*** *FLAMES Scenario Highlighter*. Plays back an executed scenario. Visualization options include 2-D and 3-D playback. FLASH can also be used simultaneously with FIRE to view the action as it is occurring.

- ***FLARE:*** *FLAMES Analysis and Reduction Environment*. Converts output data to SQL tables for mining and analysis. For simple analyses FLARE can be replaced with custom exporters that extract the necessary data using C^{++} scripts.

The interaction of the four FLAMES products is shown in Figure 156.

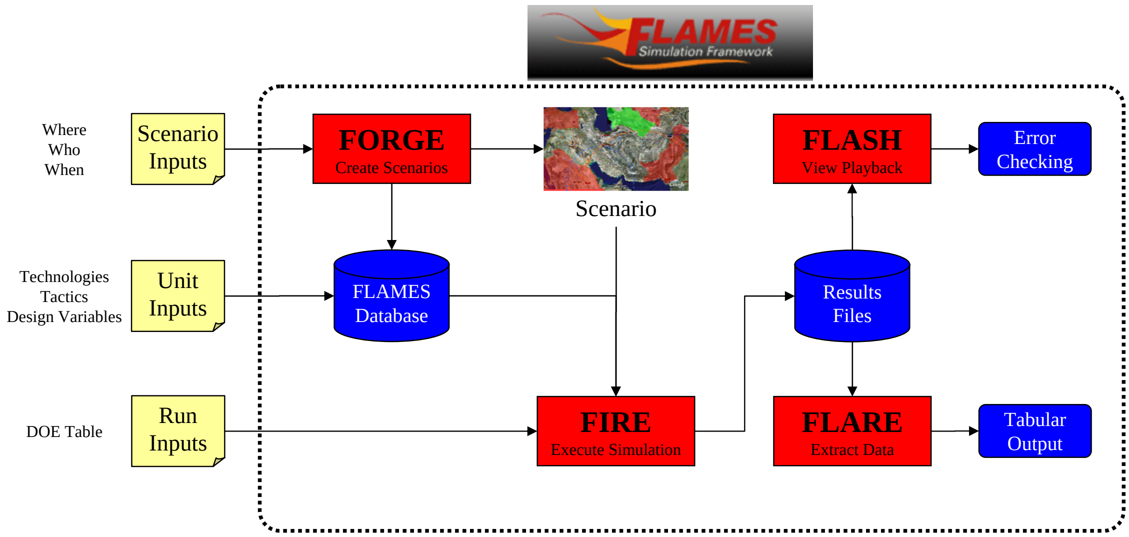


Figure 156: Information Flow in the FLAMES Framework.

To understand FLAMES, it is necessary to understand its origins. In software engineering, a framework is "a defined support structure in which another software project can be organized and developed" [16]. A framework to an engineer is his/her file cabinet, library, computer, and engineering knowhow and contains support programs, code libraries, and other software to assemble a coding project. Software developers use frameworks because the development of large scale software projects is inherently distributed. Pieces of code are written by different developers that may be geographically separated[4]. FLAMES is written by software developers and hence follows a very structured, object-oriented approach. Instead of a single monolithic entity constructed for one purpose, FLAMES is instantly reconfigurable with a variety of custom modules, interfaces, and user-defined models. Although it comes with several example models to demonstrate the functionality of the software for campaign analysis, these models are designed only to serve as starting points for custom development by the end user. In fact, the object code for the FLAMES suite of tools is provided and the code is actually recompiled when new models are created! This level of

[4]This is of course identical to the scenario faced in modern engineering design; however, the engineering community is poorly equipped to adapt since monolithic engineering firms are not as ephemeral as software development entities.

control over the functionality of a commercial product is unprecedented outside the software engineering field. It is ideal for the development of a flexible modeling and simulation environment for military campaign analysis and technology forecasting.

> ***Hypothesis 2.1****: A top-down capability-based evaluation of technologies can be performed using a holistic, object-oriented, hierarchical simulation of systems-of-systems.*

A.2 Hypothesis 2.2: Using Artificial Intelligence to Remove the Human from the Analysis Loop

One of the most severe roadblocks in the formulation of a simulation-based method for technology evaluation is the fact that most military simulations use a human-in-the-loop to make decisions. A major philosophical decision at the strategic level concerns the desire to use artificial intelligence in the form of machine learning and agent-based modeling to address this issue. Concerns related to this decision also dominates the hypotheses at the operational level.

A.2.1 Decision Making in Design and Optimization

Decision making is "the cognitive process of selecting a course of action from among multiple alternatives" [16]. This psychological construct is an important part of many professions and of every day life, especially design. According to Hazelrigg, "The selection of design parameters for an engineering system ... constitutes an allocation of resources. [Therefore,] design is a decision making process, and the selection of design parameters represent decisions" [198]. The science of decision theory, which concerns mathematics, statistics, philosophy, psychology, and economics is concerned with studying how decisions are made. Three major classes of decision making considered in this work are shown in Figure 157 and explained briefly below.

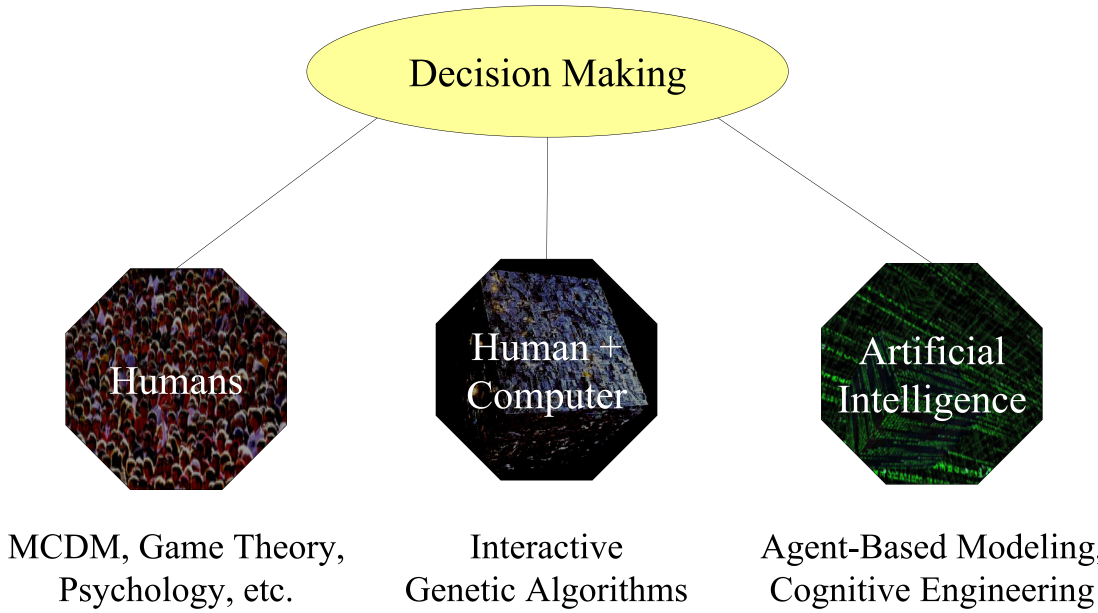


Figure 157: Three Major Categories of Decision Making.

A.2.2 Human Decision Making

Humans are very good decision makers. The cognitive processes and elements of a human brain can quickly analyze data, extract information from memory, and determine the appropriate course of action. For this reason, a human is often included in many military simulations. Generals with extensive training in tactical maneuvers, military strategy, and the capabilities of various military units are extremely efficient decision makers. For some large-scale military simulations, it is most effective to include these human decision makers: the number of simulation runs executed is often one or few, and the simulation occurs over a matter of days, sometimes synchronized with a real-time clock to mimic the chain of command in actual combat [115, 156].

Additionally, there are many different "ways" to make decisions. Multiple-criteria decision making (MCDM) is a class of techniques for making decisions "in the presence of multiple, usually conflicting, criteria" that impact the value of an alternative [210]. Decisions made under uncertainty involve probabilistic theory. Fuzzy logic, possibility theory, and game theory can also be used to make decisions under certain conditions [332]. The study of human decision making is a large area of research in the field of psychology.

A.2.3 Multiple Criteria Decision Making (MCDM)

Multiple Criteria Decision Making (MCDM) encompasses Multiple Attribute Decision Making (MADM) and Multiple Objective Decision Making (MODM). Many treatises are available on these topics including References [210] and [488]. MADM has been applied extensively to a wide range of problems including commercial transports [240], unmanned aerial vehicles [291] and ballistic missile target vehicles [60]. This research does not rely heavily on MCDM techniques for a user interface; however, a MCDM calculation using an Overall Evaluation Criterion (OEC) is used to prioritize targets within the testbed simulation. The theory behind this application is explained in Section B.1.

A.2.4 Decision Trees

A popular method of analyzing decisions is through the use of a decision tree: a graph of possible decisions and their consequences. Each node in a decision tree represents a point in time where a decision is made. Multiple branches from a node represent decisions that can be made at that time. An example of a decision tree is shown below in Figure 158. In this example, the red circle represents the first decision for which there are nine resultant paths represented by the blue circles. Each blue circle may have one or more branches. The number of pathways through this decision tree is fifty. The nodes do not line up temporally because decision nodes may be reached at different times. Also, some green nodes, representing the final state of the simulation for this small example, can be reached from multiple blue decision points while some can only be reached by one decision node.

A decision tree has limited applicability for decision making. For example, at the red node, the decision maker can choose any of the nine blue paths. Which one should be chosen? It can be argued that the blue node with the maximum utility should be chosen. Moving to this node, the decision maker now has several green nodes available; however, not all green nodes are available. By selecting the optimum blue node, the decision maker may have globally suboptimal green nodes as the available final decision pathways. As this example demonstrates, to truly find a *global optimum*, a decision maker has to search every

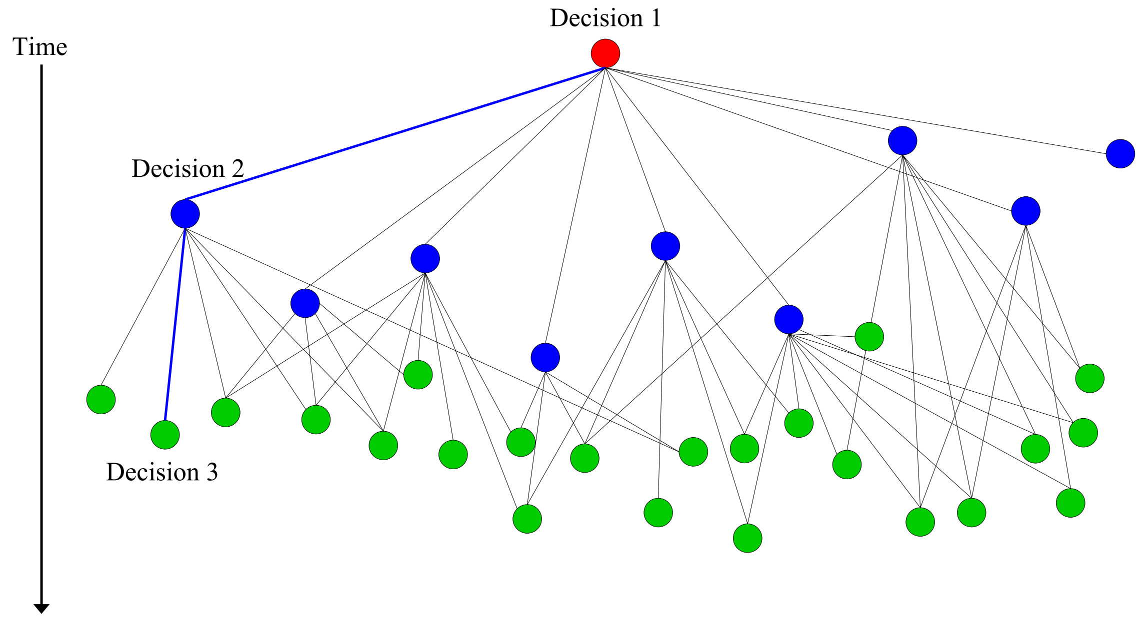


Figure 158: Decision Tree for Three Decisions at Different Points in Time.

pathway in the tree. Since the problem grows exponentially with the number of nodes and connections added, the number of pathways to explore can quickly become unmanageable. Decision trees form the concept of chess-playing supercomputers like the famous Deep Blue computer that defeated Gary Kasparov in 1997. Capable of analyzing over 200 million possible decision pathways per second, Deep Blue fathoms a decision tree "with the position on the physical chessboard acting as the root position. The moves searched become the tree branches, and the positions searched become the branching points" [209].

As is the case with the chess playing example, when an adversary is involved, the available options on the decision tree become functions of the other player's options. This is often called a *game tree* in game theory [151]. The issue of campaign simulation using decision trees is further compounded by the realization that each individual engagement can be treated as a single game, and a campaign is a sequence of many simultaneous interacting games.

Furthermore, this decision tree takes a purely deterministic view of the problem. If the value of each node is probabilistic, it becomes necessary to find the expected value of each decision before it is made. In a military simulation, this can mean that *each pathway* *must be explored thousands of times* with different values for noise parameters to assess robustness of a given pathway. For non-trivial problems, there are not enough computers in the world to discover a solution.

A.2.5 Computer Assisted Decision Making

"In many cases during the design process the analyst is interested in criteria which are either non-quantifiable or for which there are no available numerical models" [79]. To address this issue, Buonanno developed a method for using Interactive Genetic Algorithms (IGAs) to combine quantitative physics-based analyses with qualitative measures that rely on expert input. Using a graphical interface combined with a genetic algorithm, designs which score well in terms of numerical objective function are displayed to the user who ranks them as "bad, poor, ok, good, and best." The use of physics-based tools to narrow the design space is in contrast to traditional IGAs that are based solely on qualitative judgements by human operators. Takagi notes that the decision-maker's error increases with time and approaches a random process [393]. The approach developed by Buonanno could be used in a wargame if a human decision maker is used. The desire to entirely remove the human from the loop in this research drives the selection of an artificial intelligence-based approach to battle management and decision making described below.

A.2.6 Artificial Intelligence in Decision Making

A difficulty arises when constructive simulation is used to examine a wide array of concepts over larger time scales. It is not practical to sequester a thousand generals for ten years to serve as battle management decision makers as millions of code runs are executed. In fact, humans tire quickly when utilized as decision makers in repetitive interactive simulations [393]. To facilitate large design space exploration without the need for a human-in-the-loop, an approach is needed that allows a computer to serve as a surrogate decision maker. This approach was famously depicted by the War Operation Plan Response (WOPR) computer in the 1983 film *WarGames* [299].

WOPR is designed to play numerous war games, learning from the outcomes of each, and optimally respond to potential nuclear attack scenarios [16]. In the climax of the

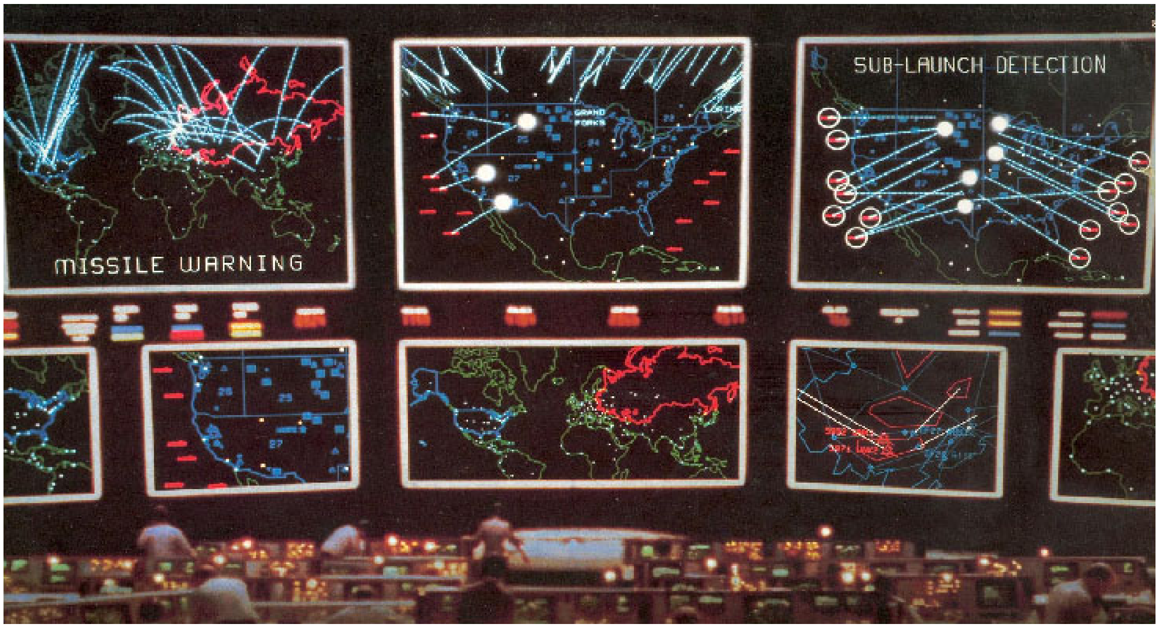


Figure 159: The War Operation Plan Response (WOPR) Computer Testing Scenarios at "NORAD" in the 1983 film *WarGames* [299].

film, WOPR plays through all possible scenarios in the game "Global Thermonuclear War" before concluding that there is no winning strategy (see Figure 159). This fictional account provides a starting point for the development of a computational cognitive decision making model.

Many treatises are available on the psychological and computational aspects of decision making, artificial intelligence (AI), machine learning, and cognitive engineering. While a fully autonomous, human-like "Meta-General" is desired, thousands of computer scientists have been at work for fifty years to develop this type of capability. This research focuses on a smaller subset of techniques to prove that AI can be incorporated to some degree in a capability-based design environment to remove the human from the loop. Instead of focusing on the detailed theory behind decision making, the purpose of this section is to provide background on several possible options for this "Meta-General" and suggest a course of action for this research.

Because of the large number of runs needed to perform a design space exploration or forecast the impact of technologies, it is not feasible to have a human in the loop to make strategic and tactical decisions. As a result, a technique that reliably approximates human decision making in an object-oriented simulation is needed. One promising approach is agent-based modeling.

A.2.7 Introduction to Agent-Based Modeling

> *"Some people worry that artificial intelligence will make us feel inferior, but then, anybody in his right mind should have an inferiority complex every time he looks at a flower."*
>
> -Alan C. Kay

Complex systems are categorized by emergent, dynamic, non-linear behavior derived from interactions between lower level components. The field of agent-based modeling and simulation (ABM/S) uses a bottom up approach to the design of complex systems that relies on creating relatively simple "agents" and defining the interactions between agents in such a way to generate realistic system level behavior with relatively unsophisticated subsystem elements. According to Ilachinski, "agent-based simulations of complex adaptive systems are predicated on the idea that the global behavior of a complex system derives entirely from the low-level interactions among its constituent agents" [212].

Through the appropriate establishment of rules, objectives, and rewards for a group of agents, some decisions can be made automatically without human interaction. "The major strength of ABM/S comes from the fact that it is a simple, versatile, and flexible method that is well suited for studies of complex non-linear systems" [258].

A.2.8 Agents

The building blocks of an agent-based simulation are called *agents*. Agents, also called actors or players, use the principle of artificial intelligence to emulate the behavior of humans. In a military simulation they represent assets available to perform actions such as tanks, aircraft, soldiers, and ships. Two main branches of focus emerge for the use of agents: design/analysis, and simulation. Design/analysis is primarily seen in the software industry and has been in development for more than a decade [222, 473]. Here, agents serve as construction tools for the development of software and software architectures. Another design and analysis example is the use of agents for mechanical assembly operations [185]. Software agents called "bots" are also used to automate certain computational tasks such as web surfing (web spiders), instant messaging (chat bots), and online shopping [16].

Definitions of the term "agent" abound in the literature:

- "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors" [363].
- "Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed" [265].
- "Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions" [197].
- "An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" [157].

As with biological organisms, agents operate in an environment and react to stimuli based on cognitive processes that seek to achieve one or more goals. Essentially, an agent can be loosely defined as "a surrogate life form." While simple agents act as surrogate life forms in the most basic sense, the term **intelligent agents** is used to extend the basic definition by adding two general characteristics of higher-order life forms, *autonomy* and *adaptiveness*. First, an intelligent agent has control over its own actions. Agents have a series of actions which they are allowed to perform as they attempt to maximize their own utility function. They autonomously interact with their environment, gathering information through sensing mechanism and making decisions based on their perception of the world around them. As the world changes, agents can also adapt and change their behavior based on what they "know" and what they perceive about their environment. Intelligent adaptive agents are inherently more realistic surrogates for human decision makers due to their ability to alter goals and beliefs based on information acquired about their environment.

The use of autonomous intelligent agents for simulation has roots in Cellular Automata which was originally developed by John von Neumann and Stanislaw Ulam in the 1940's to find a reductionist model for biological evolution [212, 213, 472]. Simulations use agents to give traits, characteristics, or personalities to individual pieces within a larger context. Using this basic approach, agent-based simulation has been used to model economics [400], the air transportation system [258, 259], endangered bird populations [65], human behavior [328], land based combat [212], and many other dynamic systems. Reference [414] contains a comprehensive survey of collectives and agent-based models and identifies over a thousand references on the topic.

Agent-based simulation has also found its way into the motion picture industry to reduce the time it takes to render detailed scenes with hundreds of "actors" and to create more challenging artificial intelligence for computer controlled opponents. Notable examples of this technology include the large 3-D rendered battle scenes in *The Lord of the Rings* [245, 317] and *Troy* [476]. Computer games have also benefited heavily from advances in agent-based simulation. To be entertaining for long periods of time, real-time strategy (RTS) games must have computer players and interactive agents that have complex behavior that varies according to user inputs. Real-time strategy games trace their origins to *Stonkers* (Imagine, 1983), *The Ancient Art of War* (Broderbund, 1984), and *SimCity* (Maxis, 1989). The RTS genre was defined by *Dune II* (Westwood, 1992), and other notable games include *Warcraft* (Blizzard, 1994), *Command and Conquer* (Westwood, 1995), *Total Annihilation* (Cavedog, 1997), and *StarCraft* (Blizzard, 1998). The recent development of Massively Multiplayer Online Games such as *Air Warrior* (Kesmai, 1987), *EverQuest* (Sony, 1999), and *Final Fantasy XI* (Square-Enix, 2002) combine hundreds of human players with intelligent agents (called non-player characters, or NPCs) in a massive, interactive, online world.

What place do video games have in scientific research or military modeling and simulation? In one example, based on the success of first-person tactical strategy games such as Tom Clancy's *Rainbow Six* (Ubisoft, 1998), and *Medal of Honor* (Electronic Arts, 1999), the United States Army announced the creation of a massively multiplayer online training

U.S. Army *AWE* — Forterra | Command and Conquer Generals — Electronic Arts | Final Fantasy XI — Square-Enix

Figure 160: Video Games Utilizing Agent-Based Simulation [178].

simulation called *AWE* (asymmetric warfare environment) to train military personnel in urban combat before they enter the combat zone [178]. Examples of some of these video games are shown in Figure 160. The level of effort in the creation of some video games is on par with the most complex military simulations. Most video games take one to three years to complete, have a staff of over a hundred developers, and cost between \$1 and \$15 million dollars to produce. According the market research firm NPD Group, PC and console hardware and software sales exceeded \$11 billion dollars in 2004 [16]. For comparison, the FY 2005 budget request for the United States Missile Defense Agency, the largest single line item in the DoD budget, was approximately \$8.8 billion dollars [322].

As an example of crossover between the entertainment and "serious gaming" industry, Breakaway Software of Hunt Valley Maryland produces entertainment titles such as *Civilization* but also develops simulation tools such as *netStrike*, *24 Blue - Flight Deck*, and the *mōsbē* simulation toolkit [6].

A.2.9 Concept of Agent-Based Modeling and Simulation

Since the use of agent-based modeling relies on the emergence of complex behavior from the lower-level interactions among constituent agents, it is necessary to review the elements of agent-based modeling and simulation as shown in Figure161. The concept relies on several elements:

1. **Goals**: What the agent is trying to do. Can include "rewards" for successfully completing objectives.

2. **Beliefs**: What the agent thinks about the world around itself according to information provided. Beliefs can be true, false, or somewhere inbetween, depending on the sophistication level of the agent logic.

3. **Information**: Data about the environment in which is is placed. Usually derived from sensory input, but can also come from communication with other agents.

4. **Senses**: The methods by which an agent can gather information about its environment.

5. **Decisions and Actions**: A series of potential decisions or actions that can be preprogrammed or discovered by the agent through a learning process.

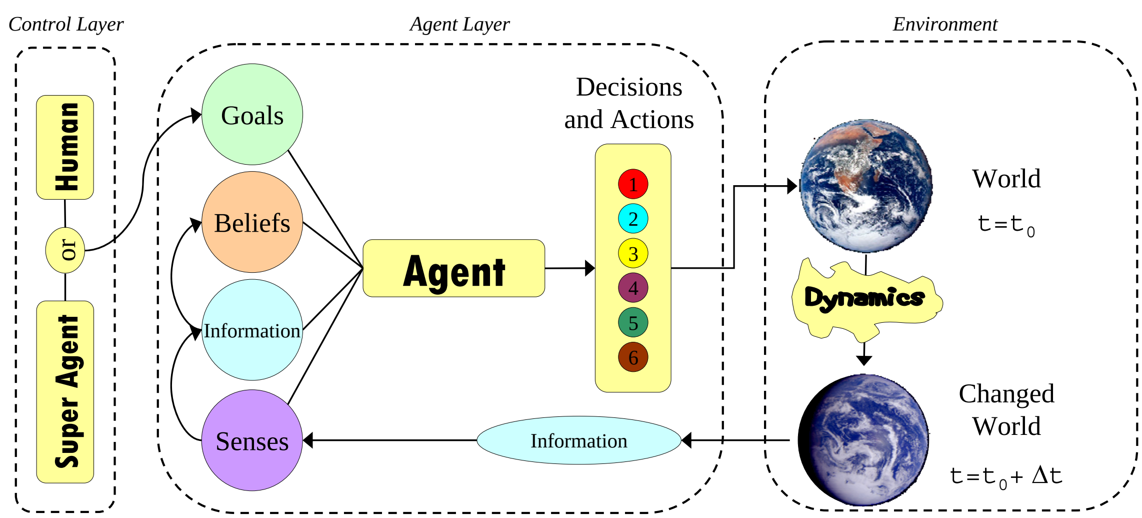


Figure 161: Concept of Agent-Based Modeling (adapted from Reference [258]).

By gathering information through its senses, an agent examines the goals and beliefs in its construct and attempts to determine which action brings it closer to its goals. This is a feedback process which involves interaction with its environment. The general concept behind agent-based modeling is that by performing actions on its environment, the agent changes that environment, albeit ever so slightly. The agent then examines the changed world through its senses, examines whether its previous decision was helpful or harmful with respect to its goals and the process repeats. The goals of an agent can be changed through interaction with a controlling super-agent or human. This is discussed in more detail in Section B.2.1.

A.2.10 The Continuum of Intelligent Agents

There are many types of agents at varying levels of complexity. A continuum of possible agents is shown in Figure 162.

Figure 162: Continuum of Agents.

The simplest types of agents are called *reactive agents*. These agents have simple rules that may be formulated as if/then statements or table lookups depending on probability. They are often deterministic and do not require a learning mechanism. Reactive agents simply respond to stimuli in a predictable manner.

At the other end of the spectrum are *deliberative agents*. Deliberative agents are highly intelligent, utilize reinforcement learning mechanisms, and sometimes can be self-reprogrammable or reconfigurable [88]. Individual assets such as fighters and SAM sites likely reside toward the left end of the spectrum, while artificial intelligence-enabled battle managers would epitomize deliberative agents. The development of highly accurate deliberative agents is an area of continuing research in computer science and mathematics.

A.2.11 Preferred Architecture for Multi-Agent Systems

Lewe describes several key categories of agent-based simulations including simulations with independent agents, an information layer, groups of agents, and a hierarchical structure [258]. The hierarchical multi-agent system (MAS) is most appropriate for military campaign modeling. This architecture is shown in Figure 163.

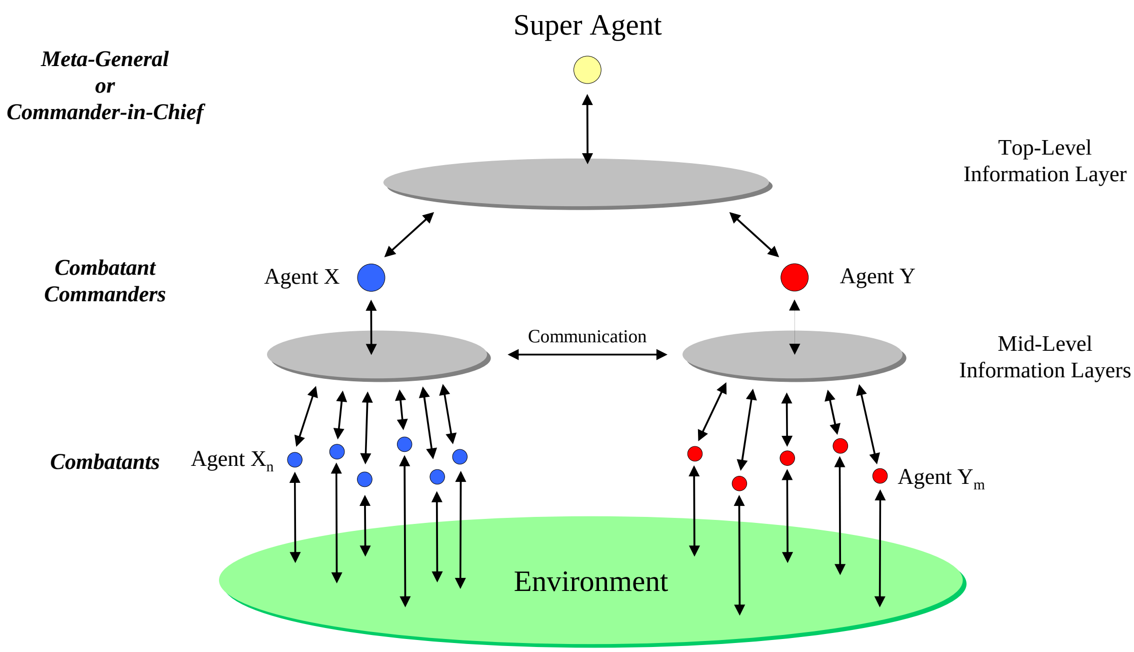


Figure 163: Multi-Agent System with Hierarchical Organization (extended from Reference [258]).

In this case, the super agent is either a human or an intelligent agent as shown in Figure 161. The super agent is representative of the general, secretary of defense, or commander-in-chief. This agent can communicate information to his combatant commanders. These commanders can exchange information and often do in large battles or joint force operations. Each commander can also communicate with the combatants under his command; however, he cannot communicate with the combatants in other units. In fact, having these extra communications pathways can result in the same agent receiving multiple and conflicting goals depending on the objectives of the agents at the combatant commander level. Finally, the lowest-level agents interact directly with the environment and pass information back up the chain of command to the super agent. In real life, this exchange of information is not lossless: information exchange takes a finite amount of time and commanders are inclined to report the more optimistic details to top-level warfighters.

This hierarchical architecture differs slightly from the pure agent-based modeling and simulation concepts outlined by Lewe [258] and Ilachinski [212]. In these frameworks, the complex behavior at the system-of-systems level is *entirely* derived from the behavior of a small number of relatively unsophisticated agents with simple rule sets. The pure agent-based paradigm holds well for large-scale systems with many similar assets, typified by land-based combat, ant populations, and ecological models. In contrast, air combat more closely relates to that of special forces or the single combat warrior due to the smaller number of highly integrated heterogeneous assets. Pure agent-based techniques are good for discovery and development of new tactics with static assets, exploring high-level behaviors and policies, and understanding the behavior of military systems under a variety of battlefield conditions. While a pure agent-based approach solves several problems related to system-of-systems, the "set-up the agents and watch them play" mentality provides too little resolution to the decision making environment to be of practical use for the selected problem of interest.

A.2.12 Paradigms for Creating Adaptive Intelligent Agents

One challenge in agent-based modeling is the determination of "how intelligent" agents should be. The WoLF ("Win or Learn Fast") approach developed by Bowling features a variable learning rate, where the agent reduces the learning rate when it is performing well (the agent becomes more "cautious" because adversarial agents may adapt) and increases the rate of learning when it is performing poorly [69]. Due to the confounding effect of agent learning on technologies, in this work intelligent agents on either side do not adapt to the patterns of the adversary.

Another question regards the scope of the problem. Casti notes that "in contrast to simple systems—like superpower conflicts, which tend to involve a small number of interacting agents—or large systems... which have a large enough collection of agents that we can use statistical means to study them " a *medium-sized* number of agents should be used to study emergent behavior [89]. This number may range from dozens to hundreds of intelligent agents.

A.2.13 Other Key Concepts

Non-linear behavior is that which is not directly proportional to the change of an input value. As Gleick eloquently notes, “nonlinearity means that the act of playing the game has a way of changing the rules” [168]. In contrast, linear relationships are proportional to the change of input parameters.

Chaotic behavior identifies random outputs of a deterministic mapping [213]. In chaos theory, the idea that small perturbations of a large dynamical system can produce large variations on the overall system was referred to by Lorenz as the *butterfly effect*[5] [262].

A **collective** is “a large system of agents where each agent has a private utility function” [414]. These agents are amalgamated together and the performance of the overall system is measured using a world utility function. The coordination of these utility functions is analogous to decomposition techniques such as collaborative optimization in the MDO community [72].

In most constructive simulations, agents are limited to **local information**, that is, no single agent has access to the information available to all other agents. Such clairvoyance is counter to the Clauswitzian notion of the “fog of war” [100]. Information is gathered through *sensors*, perceived through *cognitive models*, and distributed through *communications*.

Decentralized control means that there is “no God-like ‘oracle’ dictating what each and every part ought to be doing” [214]. Although the battlespace may have global commanders, there is no direct line of communication between non-adjacent levels of the command hierarchy shown in Figure 163. While decentralized control is a central feature in many agent-based simulations, the United States Air Force operates under a paradigm of “centralized control and decentralized execution” [431]. This doctrine is the reason why a top-level battle manager is needed to pass information to individual agents.

[5]The butterfly effect refers to a meteorological theory that the small disturbance caused by the flapping of a butterfly’s wing becomes amplified due to the chaotic nature of weather patterns to changes the atmosphere on a large scale. Lorenz postulates that long term behavior of chaotic systems is impossible to forecast.

A.2.14 Creating Intelligent Agents

There are many paradigms in the literature for creating "intelligent" agents. Several promising techniques for this class of problems are summarized in the subsequent sections.

A.2.14.1 The Brute Force Approach

Since the value of a decision is not known beforehand, one way to make decisions in the simulation environment is to simply try every possible combination of decisions using a grid search and select the best option. In an engagement or one-on-one simulation, it is trivial to simply run through all possible engagement actions and select the best one. In a larger campaign model, since every engagement depends on the results of the previous engagement, the "brute force" technique is not computationally efficient.

A.2.14.2 Adaptive Agents by Trial and Error

When the value is of a given decision is not know *a priori*, a useful technique is to allow agents to adapt to changes over time. Beginning as "naive" agents, they develop intelligence by trying different actions to see what works by using external sensors of the environment in which they operate.

While this approach has had much success in many fields, it is not entirely appropriate for this type of simulation, which is further confounded by the element of *time*. For example, on the first day of the war, the battle manager would decide to send out some F-18's to see how they do. When they all get shot down, it would try some Tomahawk missiles. The agent would likely conclude that F-18's are bad and Tomahawk missiles are good. If the agent was penalized for the cost incurred by using many Tomahawk missiles, it would keep trying F-18's every so often to see if they can penetrate enemy defenses. Meanwhile, every time it tries the F-18 solution, it *actually* loses n more F-18's because this is happening during the analytical simulation. This trial-and-error approach is (hopefully) not how the military operates. Generals do not randomly send elements into battle to see how they do: they **know** what to do. Battle management agents need to **know** how to win the war with the correct series of actions in the shortest possible time... or at least come close to it. It is impossible to always be correct in the establishment of strategy, but the ability of the U.S. military to observe the enemy, analyze intelligence, and gather data ensures that the battle management process is more reasoned than a simple trial-and-error process.

Approaches from game theory can easily calculate the optimum strategy for a given situation, but since the value of the game, dispositions of the players, and number of players are changing over time as the game is continually played, the selection of an optimum strategy through trial and error is not appropriate for this class of problems.

A.2.14.3 Knowledge-Based Systems and Machine Learning

A knowledge-based system (KBS) is useful for querying a database of information. This is usually done by asking a series of questions that go through a process of elimination to narrow the search space in the database to one answer. A major drawback to knowledge based systems is the amount of time needed to populate the database. Also, in most cases, extensive rule-sets must be specified for how to made decisions regarding the options in the database. Furthermore, the rule sets are usually hardcoded for what was in the database when the rules were written. Adding new information necessitates a redesign of the decision making algorithm.

Despite these drawbacks, simple "if-then" type rule sets are adequate for the cognitive models of simple reactive agents. In an object-oriented framework, several generic cognitive models can be created that describe basic responses to specific stimuli. As previously mentioned, reactive agents need not be adaptive and intelligent provided that their rule set sufficiently incorporates the necessary behaviors of interest.

Deliberative agents such a the battle manager are not well suited for static knowledge-based systems; however, a modification to a dynamic and adaptive KBS is of interest. An entertaining example of an adaptive knowledge-based systems is the neural network-based 20Q handheld game [1]. The game is played by thinking of something and answering questions posed by the user interface. An example of the question sequence for "a fighter plane" is shown below in Figure 164.

20Q has assembled its knowledge base from responses of game players over the past

Q17. I am guessing that it is a fighter jet?
Right, **Wrong**, **Close**

16. Is it used by the police? **No**.
15. Can you use it with your friends? **No**.
14. Can it save your life? **Yes**.
13. Is it commonly used? **Yes**.
12. Can you buy it? **Depends**.
11. Is it located in the sky? **Yes**.
10. Can it be used for recreation? **Sometimes**.
9. Can you live in it? **Depends**.
8. Can it be refilled? **Yes**.
7. Does it travel on roads? **No**.
6. Does it move? **Yes**.
5. Is it worth a lot of money? **Yes**.
4. Is it smaller than a loaf of bread? **No**.
3. Is it soft? **No**.
2. Do you use it at work? **No**.
1. It is classified as **Other**.

Figure 164: 20Q Online Game, Results for "Fighter Plane."

nineteen years [1]. As time goes on and more people provide it with more information, the connectionist algorithms in the routine add more information to the database and more closely approximate the human decision making process. This adaptability is of great interest in the development of an intelligent battle management agent. The field of machine learning is interested in circumventing the drawbacks of traditional knowledge-based systems to have the system "discover" new knowledge and solutions through continuous feedback.

Ilachinski notes that there are four major categories of machine learning techniques[6] [213]:

1. **Analytic Learning**: Analytic learning systems require a thorough understanding of the general underlying problem type and must have available a large number of problem-solution exemplars. The technique relies on adapting solutions to problems that it identifies as being "close to" known solutions to known problems.

2. **Inductive Learning**: Inductive learning requires an external "teacher" to produce problem samples. The teacher grades the system's attempts to use its stored knowledge to try and solve each problem in turn. The teacher's grade is then used to update the system's knowledge.

[6] The bulleted list is directly quoted from Ilachinski (2004), page 198.

3. **Neural Network Learning**: The neural net (also called the Connectionist) approach consists of applying a learning algorithm (such as back-propagation) to adjust a set of internal weights in order to minimize the "distance" between calculated and desired solutions to selected problems. Given a set of training problem-solution exemplars, the learning algorithm produces a network that, in time, is able to correctly recognize the pattern implicit in all input (i.e. problem) and output (i.e. solution) pairs.

4. **Genetic Algorithm (or Selectionist) Learning**: Selectionist learning systems exploit the learning capability of a genetic algorithm to "evolve" an appropriate knowledge base.

While Ilachinksi notes that the most appropriate method for decision making is the genetic algorithm, this is merely a more sophisticated "trial and error" approach (see Section A.2.14.2). In the military simulation, the amount of trial and error should be reduced as much as possible as "failed" cases contribute to abysmal scores in terms of measures of effectiveness. If the battle manager is not a strategic genius, it is difficult to distinguish between the time delay for the battle manager to "get smart" and the relative effectiveness of a given technology. An technique that uses one or more of the above approaches is needed.

A.2.14.4 The Tactical Genius

The ultimate in battle management is a brilliant tactician that knows exactly what to do in every situation; however, the question is posed, "do such humans even exist?" The asymmetric advantage of modern technology greatly increases the effectiveness of each sortie but reduces the opportunity to gain combat experience. In Operation *Iraqi Freedom*, USAF bombers flew 505 out of 41,404 sorties (1.2%) and only eleven B-1, four B-2, and 28 B-52 bombers were used in the 28 day conflict [304]. Developing a database of tactical information based on the limited experience of active duty pilots in a low-threat environment using a knowledge-based system may not yield accurate results against different types of threats.

Constructive simulation can be used to develop a computer-based tactical genius, similar to the training of the computer in the climax of the 1983 film WarGames (see Section A.2.6).

One potential approach is to create at random a large number of notional missions, randomly select equipment to be used on each mission based on high-level parameters for each type of equipment, fly the sample missions, and evaluate the outcome of each. After a large enough sample of missions is performed, the battle manager can obtain sufficient knowledge to infer what should be done in similar situations. As described by Hawkins, "There will be no need or opportunity for anyone to program in the rules of the world, databases, factors, or any of the high-level concepts that are the bane of artificial intelligence. The intelligent machine must learn via observation of its world" [195].

This machine learning approach is closest to the aforementioned *analytic learning* technique though it may use elements of the neural network or genetic algorithm approach to provide adaptivity. An application of this approach is summarized in Section 5.5.

> ***Hypothesis 2.2:*** *Techniques from machine learning and agent-based modeling can be leveraged to provide an intelligent battle manager with an "understanding" of basic strategic and tactical decisions.*

A.3 Hypothesis 2.3: A Comparison of Optimum and Robust Technology Portfolios

Optimization, literally the pursuit of that which is best, is inherent in every process in our daily lives. The term optimization generally connotes an iterative analysis process and the mathematical techniques used therefore [470]. In engineering, optimization "is traditionally thought of as weight minimization, ignoring consideration of other attributes" [406]. The ultimate "optimization" of an LRS system architecture and its accompanying portfolio of technologies relies on the selection of an appropriate objective function that considers the multidimensional tradeoffs endemic to this problem.

Despite recent trends in the literature, optimization of systems-of-systems should not merely be seen as an extension of techniques developed in the multidisciplinary optimization community [44, 111, 237]. Numerical optimization is extremely difficult with the large numbers of variables and discontinuous behaviors present in this class of problems. It is well known that decomposition of a highly complex problem and optimization of the subproblems often produces globally suboptimal results. Also, numerical optimization techniques for multiple objectives, typical of military problems with multiple MoEs, are difficult to set up and validate [122]. Most importantly, because systems-of-systems are seldom operated at a single "design point" and since the actual operating conditions for a system are difficult to forecast many years ahead of time, optimization of a technology portfolio for a single design mission is not appropriate.

The term **robust** is used to describe "a system that has demonstrated an ability to recover gracefully from the whole range of exceptional inputs and situations in a given environment" [22]. It is postulated that a more useful objective for a technology-evaluation methodology is a portfolio that is robust to changing operational conditions, evolving threats, and multiple scenarios is more useful for long term planning.

Air Force policy statements support this observation. In 2004, General Jumper stated that the Air Force leadership recognizes "that operational shortfalls exist early in the kill chain and are applying technologies to fill those gaps" noting that a robust portfolio is needed to address these shortfalls [29]. Echoing these comments, Davis notes that capabilities must be "flexible, adaptive and robust" [120]. Finally, Titus states that the goal of capabilities-based planning "is to plan for robust, flexible forces, capable of meeting a wide variety of threats, rather than an 'optimal' force for a narrow set of threats" [412].

These comments, made by experts in the field, support the assertion that a robust portfolio is more useful than one optimized for a point solution. The implementation of this desire into a simulation-based quantitative technology evaluation environment may require an iterative approach to procedural optimization. As noted by Noble and Tanchoco, "the procedural approach is a trade-off process where the objective is modified as the design proceeds. The solution that results is the solution that satisfies all the design objectives in the best manner" [321]. Since new capabilities and objectives evolve rapidly throughout the life-cycle of the program, the design too must evolve to meet these changing needs.

> ***Hypothesis 2.3:*** *For systems-of-systems, determination of a technology portfolio that is robust to changing threats and variable operating conditions is more useful than a portfolio optimized for maximum effectiveness.*

APPENDIX B

REVIEW OF OPERATIONAL HYPOTHESES

Research Question 2, "How can military simulation runs be executed without a human in the loop to make strategic and tactical decisions?" is the critical issue associated with simulation-based analysis of technologies. Simulation is often dominated by human decision makers that tire quickly when asked to analyze a myriad of repetitive situations [393]. In Section A.2, artificial intelligence was identified to address this research question. The next two hypotheses explore the AI paradigm in more detail and identify new methods for using machine learning and intelligent agents to emulate human decisions and enable rapid simulation-based technology evaluation across a large design space.

B.1 *Hypothesis 3.1: Applying QFD and MADM to Target Prioritization*

Research into systems engineering techniques and multiple-attribute decision making methodologies can be synthesized into a method for prioritizing targets based on the relationships between the strategic objectives and the target set(s) to which the target belongs. The Quality Function Deployment (QFD) technique is a systematic mathematical process used to translate the "Voice of the Customer" into the "Voice of the Engineer." Developed by Dr. Yoji Akao and Dr. Shigeru Mizuno in the 1960's, QFD has found widespread application in the development of products for the aerospace, automotive, electronics, and other industries[1]. According to Dieter, a survey of 150 companies taken in the late 1990's indicated that 71% of these companies had adopted QFD in that decade. Of these companies, 83% believed that the QFD tool "increased customer satisfaction with their products, and 76% felt it facilitated rational design decisions" [135]. Quality Function Deployment is also commonly known as the House of Quality due to the trademark shape of the matrix used to

[1] A primer on the QFD methodology can be found in the INCOSE Systems Engineering Handbook [217].

perform the method, as shown in Figure 165. The term “deployment” refers to the ability to deploy the results of this matrix to lower levels, where the engineering characteristics or“how’s” in the first matrix become the “what’s” for the lower tier.

While the traditional formulation for the QFD matrix is to translate customer requirements into engineering characteristics, the same “transfer function” can be applied to strategic objectives and target sets[2]. In this formulation, all of the “rooms” of the QFD are not used, rather, only the main room of the QFD called the interrelationship matrix is of interest, as shown in Figure 166.

Using the relative importance of a customer requirement and the interrelationship matrix that relates the customer requirements to the engineering characteristics, the relative importance of each engineering characteristic can be calculated using Equation 20. This technique is extended to the calculation of the relative importance of each target set with respect to top-level strategic objectives in Section 5.4.1.

$$(\text{EngineeringCharacteristic})_j = \prod_{i=1}^{n} (\text{CustomerRequirement})_i \, (\text{Interrelationship})_{ij} \qquad (20)$$

While many techniques for multiple-attribute decision making abound in the literature, the simplest and most effective approach to this problem is the use of an overall evaluation criterion (OEC). An OEC encapsulates multiple objective functions into a single numerical index that enables comparison of two or more products or processes. Examples of overall evaluation criteria include grade point averages, rankings in figure skating, or the “star” ratings given to motion pictures. In the case of these OEC’s, the metrics of measurement are identical and the basic OEC is simply the average of the different scores. In other cases, dissimilar metrics such as speed, weight, and range may be used requiring a normalization of the parameters as shown in Equation 21. In this case, parameters that should be maximized are scaled with respect to the baseline in the denominator while the opposite is true of metrics like cost and weight that should be minimized. Greek letters indicate the relative importance of each metric and can be scaled to represent different customer preferences.

$$OEC = \alpha \frac{Range}{Range_{Baseline}} + \beta \frac{Speed}{Speed_{Baseline}} + \gamma \frac{Weight_{Baseline}}{Weight} + \delta \frac{Cost_{Baseline}}{Cost} \qquad (21)$$

[2] A demonstration of this technique applied to the Iraq scenario is shown in Section 5.4.1.

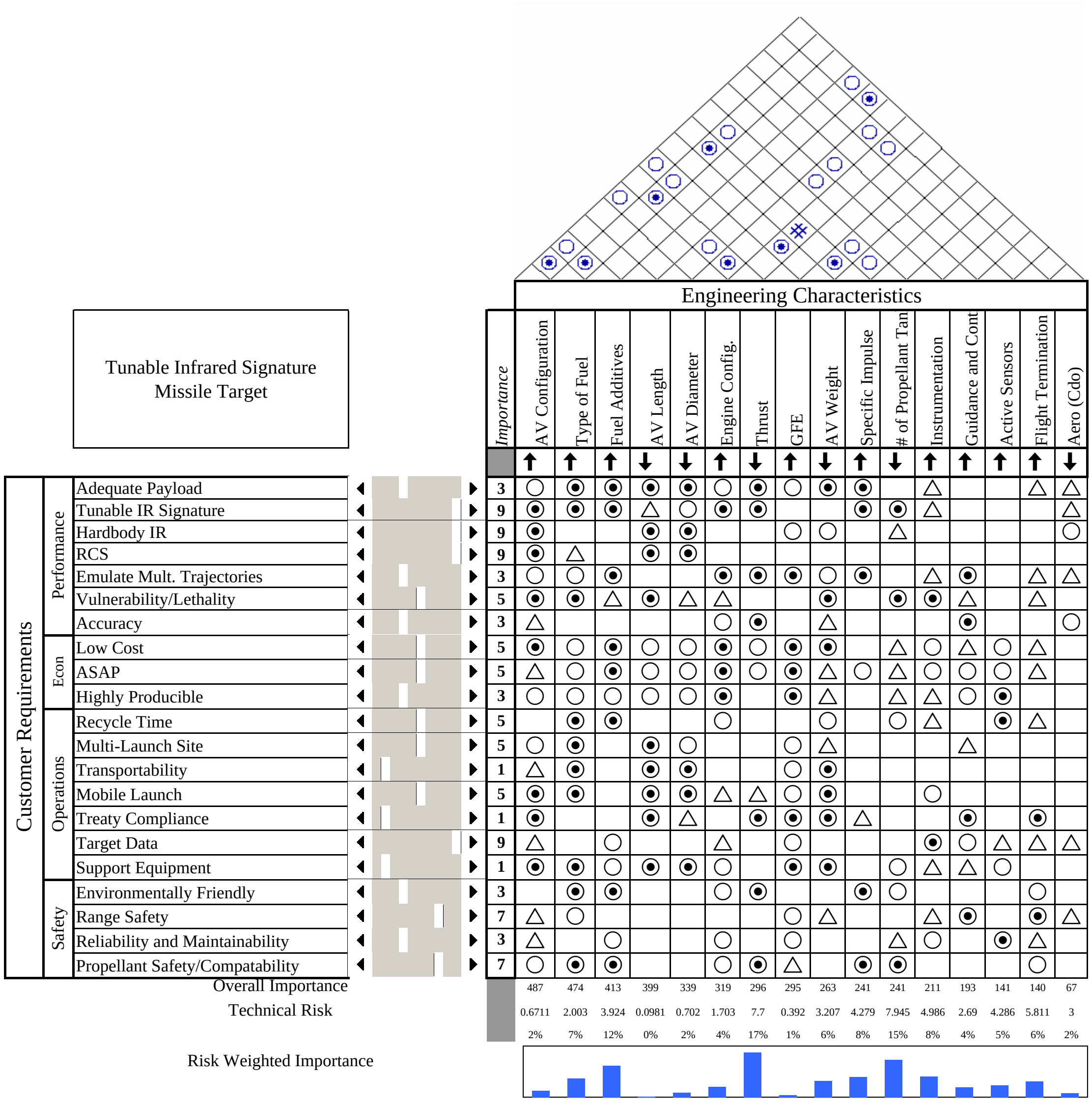


Figure 165: Quality Function Deployment Example for a Long Range Liquid Booster Target Vehicle [63].

		Engineering Characteristics		
		Part Count	Specific Impulse	Motor Weight
Customer Requirements	Importance	Min	Max	Nominal
Long Range	3		9	9
Low Cost	9	3		9
High Reliability	9	9		3
		108	27	135

Figure 166: Using the QFD Interrelationship Matrix to Map Customer Requirements to Engineering Characteristics.

Target prioritization uses both of the above concepts. The OEC technique is used to calculate the relative "threat level" of a target across twelve target sets. The QFD technique is used to calculate the relative importance of each of the target sets with respect to the customer's strategic objectives, scaling the OEC to reflect different priorities in the scenario.

> ***Hypothesis 3.1****: Quality Function Deployment (QFD) and Multi-Attribute Decision Making (MADM) techniques can be used to prioritize targets based on desired strategic objectives.*

B.2 Hypothesis 3.2: Using Intelligent Super-Agents to Create an Intelligent Battle Manager

In addition to the myriad of requisite inputs, military campaign codes usually necessitate a human-in-the-loop to examine the condition of the battlefield and direct assets [228]. Automated simulations have traditionally been constrained by hard-coded rule sets that only permit certain actions. Simulations are usually executed on a grand scale and are essentially computerized sand-table games to evaluate force level effectiveness.

Soban overcame the human-in-the-loop problem through the use of decision trees. While this technique was used to demonstrate the validity of the PoSSEM methodology, Soban notes that it is impractical for extremely large simulations as the number of cases that must be run to generate discrete sets of response surface equations for every branch becomes unmanageable [378].

Both Frits and Soban examined the implication of tactics in a modeling and simulation environment; however, the modeling techniques of the time permitted only one or two variables to be changed [159, 378]. To incorporate and objective function that encompasses multiple MoEs and develop a robust simulation capability that adapts to changing conditions, decisions must be made as the campaign progresses and the snapshot of the scenario changes in time[3].

The problem of how to incorporate changing tactics and situations in the design process

[3]Section A.2.1 summarizes the decision making process in more detail and Section A.2.7 introduces the concept of agent-based modeling and simulation, which is an enabling technology for this research.

is a daunting one. While the human-in-the-loop solution is valid for small numbers of runs, it is not practical for the generation of response surface equations. Also, the lack of reproducibility with human decision making invalidates many of the statistical techniques required for a rigorous study of the problem [393].

A promising technique that takes advantage of advances is the fields of computer science and machine learning is the use of artificial intelligence to simulate human decision making. In Section 3.4.5, the assertion was made that "techniques from machine learning and agent-based modeling can be leveraged to provide an intelligent battle manager with an 'understanding' of basic strategic and tactical decisions." A proposed approach to leverage this technique is addressed in the subsequent section.

B.2.1 Creating an Intelligent Battle Manager

> *"If the machines could just talk to each other, we would know [where the launcher is, what weapon is available to take it out, where the missile will hit] instantly."*
>
> -General John P. Jumper,
> Regarding "Predictive Battlespace Awareness" [338]

Military decision making is driven by generals and battle managers who analyze the battlespace and make decisions on how to task available units into missions. The generals make these decisions based on their training and experience. In the absence of well-trained generals or military tacticians to make decisions in an interactive simulation, can intelligent agents be developed that mimic the actions of a general?

Experience is "active participation in events or activities, leading to the accumulation of knowledge or skill" [22]. It is hypothesized that if an agent-based battle manager or "Meta-General" can be provided with enough "experience," it can make intelligent decisions. Since a campaign is made up of many similar types of engagements, one approach would be to execute a simulation in two modes: training and analysis. In the training mode, an algorithm would generate a large number of engagements at random with a variable platform and threat. Strategy and supporting tactics would be also be created stochastically and the outcome of the choices made against the threats provided would be assessed and assigned a measure of effectiveness. In the analysis mode, the Meta-General observes a situation on the battlefield and uses a table-lookup to find the best available solution to address the threat based on its prior experience from the training mode. These two methods of using a constructive simulation are depicted graphically in Figure 167.

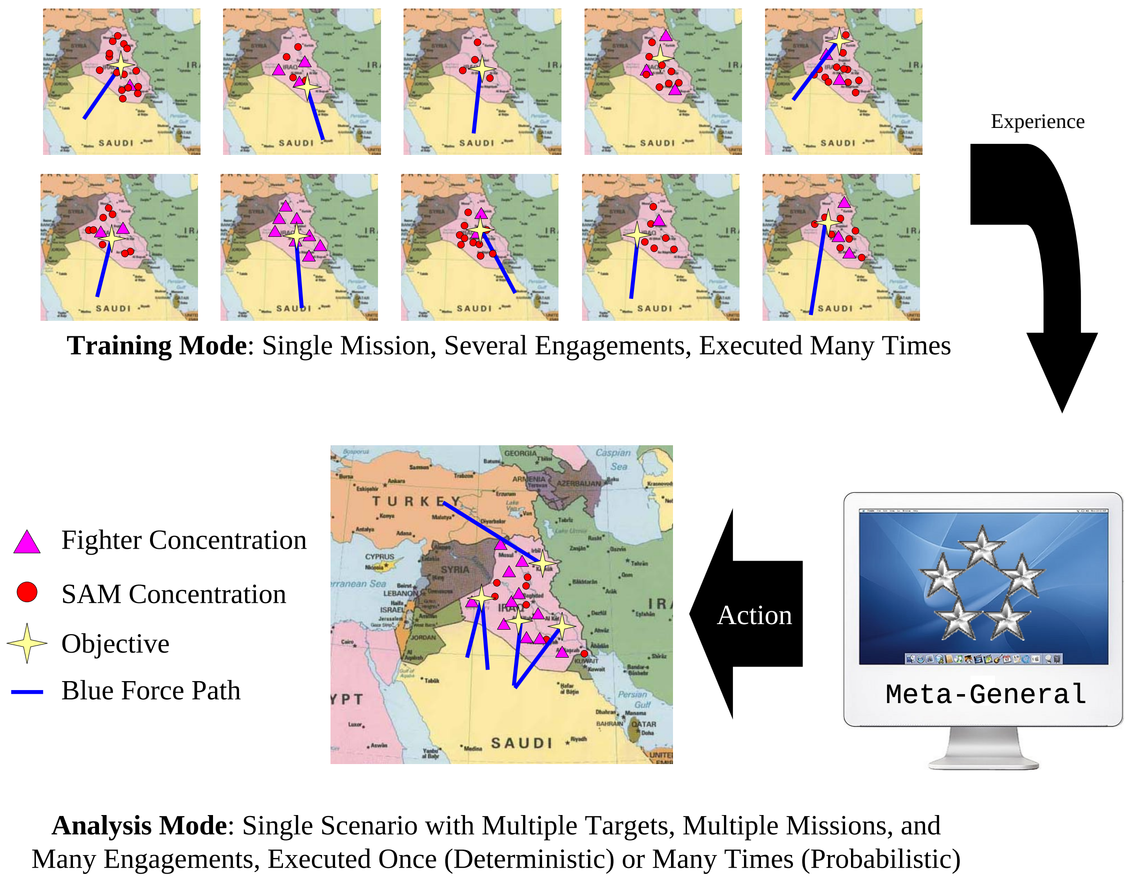


Figure 167: Two "Modes" of Operation, Training and Analysis.

Since table lookup algorithms for large, multi-dimensional tables are slow, **the preferred approach is to use neural network surrogate models to regress the experience information from the training mode**. This requires more up-front setup but allows the exploratory simulation to run much faster. The result is an equation that tells the Meta-General what course of action will be most successful based on an assessment of the threat environment.

B.2.2 Decision Making with the Meta-General

At distinct intervals during the simulation, the Meta-General fuses information from available sensors with which it can communicate to obtain a common operating picture that describes the state of the battlefield. With this information, the Meta-General uses the neural network equations mentioned in the previous section to compare the situation it observes with other situations it has been trained to recognize. For each target in the existing target list, the Meta-General iterates through available platform/weapon pairings and evaluates the neural network equations to identify those that have the highest probability of success against the detected targets under the conditions specified. The pairings are constrained by asset availability and rules of engagement. As hostile assets are eliminated from the battlefield, the Meta-General continues to observe a changing situation and alters its strategy according to the training it has received. To avoid confounding between the Meta-General's knowledge and technology effectiveness, the Meta-General does not learn and adapt in analysis mode.

In summary, the rationale for the selection of strike packages is defined by the amount of training the Meta-General receives for a range of threats, the spectrum of capability within the hostile country, blue base and asset availability, tactics, and available technology levels. Using intelligent super-agents, the human can be removed from the analysis pathway, performing only the initial setup, training, and validation of the Meta-General. This addresses the need to remove the human from the loop when executing hundreds or thousands of cases using constructive simulation.

B.2.3 Drawbacks to the Meta-General Training Approach

Unfortunately, to train the Meta-General, it is necessary to encapsulate a number of rules within the simulation to reduce the number of variables necessary for the neural network regression. Large number of variables also make validation of the Meta-General's behavior cumbersome. If fundamental changes are made to the cognitive models of systems in the architecture (for example, the physics of how airplanes fly or the heuristics involved in air-to-air engagement), the Meta-General has to be retrained. If additional "plays" are added to the playbook, the Meta-General can rely on his previous training for the old plays and simply receive supplemental training on new plays. However, if the simulation code or underlying models are fundamentally changed, the Meta-General must be re-trained from scratch. Depending on the complexity of engagements, this could be a computationally expensive process.

> ***Hypothesis 3.2****: An "intelligent battle manager" created using agent-based modeling, machine learning techniques, and surrogate models can remove the confounding effect of tactics on technology evaluation.*

B.3 Hypothesis 3.3: Using Surrogate Models to Provide Intelligence to Individual Agents

A difficult problem in a system-of-systems is how to exploit technologies to show a measurable benefit in the presence of a myriad of interacting factors that can confound the impact of technologies. Agent-based modeling (See Section A.2.7) is an technique that can be leveraged to address this challenge.

While the cognitive battle manager or Meta-General can be trained to develop a strategic battle plan, Air Force Doctrine notes that it is inappropriate for air tasking authorities to micromanage the tactics and operation of individual assets [177, 431]. After receiving the Air Tasking Order (ATO) that directs platforms to targets, individual assets (agents) carries out their orders by following a predefined series of rules that are often quite simple. Cognition models that define this rule set are exercised to plan an optimum route to the target, monitor fuel and weapon loads, and determine a course of action if the agent is engaged enroute to its target. Since the outcome of the mission depends on the world around the agent, a question arises: *what if the agents can forecast what will happen in the future and their behavior can be tuned to maximize the probability of mission success*?

While most cognition models use hard-coded rule sets whose paths are activated when discrete events occur, an intelligent agent can "predict" potential future actions if it has enough information about possible situations that may arise during the mission. One way of providing an intelligent forecasting capability is the use of surrogate models to evaluate performance for a variety of different conditions. Inputs to the surrogate model include:

- What the agent knows about itself through its own state vector.
- Information about the surrounding environment gleaned through sensors on the agent.
- k-factors on agent-level performance representing the infusion of technologies.

If any of the three input classes are altered, the agent is driven to alter its cognitive path based on the predicted performance as calculated by the surrogate model. For example, when a blue fighter detects a red fighter in the vicinity, it must decide to engage or escape. The first static rule set determines whether or not the agent has air-to-air weapons. If such weapons are available, the next decision depends on the agent's belief about its capability relative to an adversary.

The decision on whether or not to engage is therefore a function of an agent's perception of its own performance and can be defined in terms of a Performance Vector of Attributes (PVA). An example of some performance metrics that can be included in a PVA are top speed, turn rate, radar cross section, and other vehicle level attributes. These metrics are a function of the agent's current state vector: altitude, speed, drag coefficient (related to the amount of external weapons and fuel tanks), g loads, etc. Since the PVA is simply a mathematical relationship that calculates overall performance a function of the agent's physics-based parameters and the operational environment, a surrogate model can be constructed to rapidly approximate this mathematical relationship over a wide range of operating conditions.

In some cases, the agent's decisions also depend on the PVA of a potential adversary. In this case, the agent can calculate its PVA and relate it to the estimated PVA of the enemy. Since it is not possible to know the exact state of the adversary, some uncertainty surrounds the determination of its PVA. The two PVA's can be compared and the agent decides whether or not to engage the adversary:

$$\Delta PVA = PVA_{(Blue)} - PVA_{(Red)} \tag{22}$$

The threshold on the allowable ΔPVA is defined by the doctrine settings. The amount of "intelligence" about its surroundings and the knowledge provided to a given agent can be defined by the user.

Both polynomial and neural network equations can be used to provide intelligence and tunability to assets using this technique depending on the complexity of the behavior to be forecast. As an example, two F-15 aircraft can be placed side by side. Aircraft A is enabled with intelligence from surrogates and Aircraft B is not. If an engine technology is applied to both aircraft, both agents may see improved TSFC or thrust-to-weight ratio. Using a surrogate model, Aircraft A is able to relate how the new technology impacts its abilities because the surrogate relates the change in TSFC to a change in total vehicle performance. Using surrogates to provide awareness and intelligence to the agents is more representative of a real-world environment where pilots are trained to exploit new technologies as opposed to executing random missions with no knowledge of how the aircraft has been impacted by technology infusion.

This approach differs slightly from an adaptive neural network-based approach because the performance space for a simple asset such as an aircraft can be calculated as a function of vehicle attributes *a priori* and surrogate models can be generated. A central tenet of agent-based modeling is that the simple interactions between intelligent agents reveal complex emergent behavior and dynamics that cannot be easily programmed. This postulate can be used to allow asset-level agents to exploit the benefits of technology by using surrogate models to provide intelligent forecasting to "simple" individual agents whose combined interactions produce an overall desired effect that can be measured against capability-level MoEs.

To test whether this technique can be practically applied, surrogate models were generated for an aircraft using an energy-based formulation for the calculation of thrust to weight ratio advocated by Mattingly [278]. This equation is of the form:

$$\frac{T}{W} = \frac{qS}{\beta W_{TO}}\left[K_1\left(\frac{n\beta}{q}\frac{W_{TO}}{S}\right)^2 + K_2\left(\frac{n\beta}{q}\frac{W_{TO}}{S}\right) + C_{D_o}\right] + \frac{1}{V}\frac{d}{dt}\left(h + \frac{V^2}{2g_o}\right) \qquad (23)$$

Where: T/W is the thrust to weight ratio at an instant in time

q is the dynamic pressure

S is the wing area

β is the weight fraction (W/W_{TO})

W_{TO} is the takeoff gross weight

K_1 is the coefficient for drag due to lift

K_2 is the profile drag due to other factors

n is the instantaneous load factor in g's

C_{D_0} is the zero lift drag coefficient

V is the velocity in ft/sec

h is the altitude in ft

and g_0 is the acceleration due to gravity

Four surrogate models were created for top speed, climb, instantaneous turn, and horizontal acceleration. Using the above variables as inputs, an aircraft can calculate the thrust/weight ratio required to perform each of the four activities at a given instant in time. With an engine deck and knowledge of its current weight, the aircraft agent can also calculate its current thrust/weight available. The difference between the available and required values is related to the specific excess power of the aircraft in each of the four activities. The instantaneous performance vector of attributes for the aircraft can therefore be calculated as:

$$PVA = \alpha\,(\Delta SpeedPower) + \beta\,(\Delta TurnPower) + \gamma\,(\Delta ClimbPower) + \delta\,(\Delta AccelPower)\,... \tag{24}$$

Where the weight coefficients are threat dependent. For example, if the nearby threat is an air-to-air fighter, turn rate may be more important than horizontal acceleration whereas a nearby SAM threat may prioritize horizontal acceleration and top speed in the PVA. In this manner, agents can exploit technologies through an increase PVA which defines their proclivity to engage a nearby target. Technologies impact the components of the PVA and a positive difference in PVA relative to an adversary may cause an agent to engage. The decision on the threshold of ΔPVA is governed by doctrine. An example of the surrogate models used for the speed category of the PVA is shown in Figure 168.

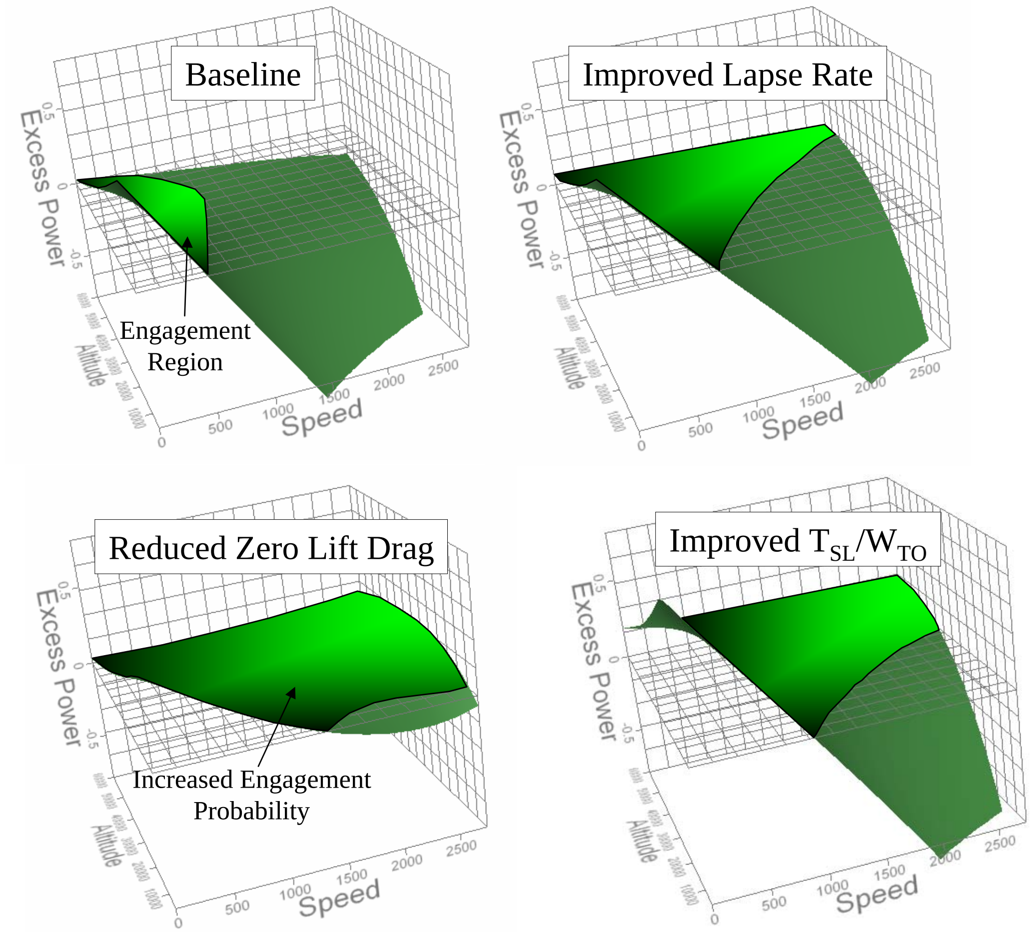


Figure 168: Example of Surrogate Models to Provide Intelligence and Tunability to Agents.

The upper left contour profiler represents the baseline configuration mapped against flight speed and altitude. The bright green surface above the grey mesh at zero indicates the flight envelope for which the agent has an advantage in top speed (holding all other settings constant). If a technology is infused that improves lapse rate, as shown in the upper right corner of the figure, the bright green area above the grey meshed area increases, meaning that the agent is inclined to engage the enemy at a greater speed and lower altitude. The same trends are evident for technologies that reduce zero lift drag and improve the sea level thrust-to-weight ratio. Using this surrogate model, the agent can tell *how much better* it is in various dimensions due to the application of technology. Similar surrogates can be provided to explain the capability of its weapons, avionics, self-protection, and ally performance. These surrogate models are aggregated together into a PVA which is used to make decisions within the cognitive logic. Similarly, surrogate models can be created using the energy-based formulation for adversary aircraft, although the PVA of hostile assets is bounded by some uncertainty since the friendly agent cannot know everything about the adversary's state vector. A series of surrogate models can be combined with cognitive models to "tune" the agents to better utilize technologies *and* advanced tactics to accomplish their mission.

B.3.1 Validity of the Proposed Approach to Battle Management

As military organizations move toward network centric operations the capabilities provided by computers and communications offer the possibility of consolidating complete battlespace control under a single commander. This research proposes an approach that utilizes advances provided by network centric technologies; however, *does not advocate that the battle manager directly control the actions of individual agents*. This approach is consistent with Air Force doctrine: "Centralized control[4] and decentralized execution of air and space power are critical to effective employment of air and space power" [431]. Under this paradigm, the command and control elements "maintain a broad theater perspective in prioritizing the use of limited air and space assets to attain established objectives in any contingency across the range of operations[5]" [431]. According to Operation *Desert Storm* Joint Forces Air Component Commander (JFACC) Charles Horner, the "unqualified success of [the] air campaign validates [the] concept" of centralized control by allowing the most effective use of all available assets and avoiding duplication of effort [208].

[4]It is important to note that decentralized control "can lead to actions being taken that may conflict with one another" [41].

[5]This paradigm originated in the 1943 version of the War Department Field Manual 100-20, Command and Employment of Air Power: "The inherent flexibility of air power is its greatest asset... control of available air power must be centralized and command must be exercised through the air force commander if this inherent flexibility and ability to deliver a decisive blow are to be fully exploited" [459].

Due to the inherent flexibility and versatility of aerospace forces, "decentralized execution permits the flexibility to maximize tactical success" [430]. The execution is decentralized through the delegation of the authority to perform the required actions to capable lower level commanders and airmen. While "modern communications technology provides a temptation towards increasing centralized execution of air and space power," history demonstrates that removing tactical authority from local commanders can have disastrous results [98, 177, 431].

The proposed approach of using an intelligent "Meta-General" for battle management (Section B.2.1) and tunable cognition models for execution (Section B.3) maximizes flexibility and versatility while avoiding rigid, predictable actions. Furthermore, the identification and development of new tactics that exploit technologies would not be possible using a fully centralized control and execution approach.

> ***Hypothesis 3.3****: At the agent level, intelligence can be provided by surrogate models and tunable cognition models that allow the agent to forecast future decisions based on the effect of technology on system-level measures of performance.*

APPENDIX C

REVIEW OF TACTICAL HYPOTHESES

C.1 Hypothesis 4.1: Physics-Based Models are Most Appropriate

A model is a "a simplified description of a complex entity or process" [22]. Since all models are approximations, the statement by Box and Hunter that "all models are wrong; some models are useful" is apt [71]. Nevertheless, since resource availability limits the amount of physical testing for systems-of-systems, models are needed to understand the underlying behavior and phenomenology of actual systems. A popular type of model is an empirical model, which is based on a regression of historical data. This type of model is extremely accurate for the development of concepts that are similar to the database from which the model is derived; however, they perform poorly for extrapolations beyond the state-of-the-art. The rule-of-thumb equations in Roskam's series on airplane design are examples of empirical models [356].

In contrast, physics-based models are derived based on physical principles governing the actual performance of systems. While simplifying assumptions such as "rigid body" or "point mass" are often used, physics-based models can capture the behavior of systems for which no historical data is available. Examples of physics-based models include the Breguet Range Equation, free-body diagrams, equations of motion, and the energy-based sizing formulation defined by Mattingly [278].

A hybrid (or semi-empirical) approach is typical of many synthesis and sizing codes such as FLOPS and ACSYNT [40]. These codes use physics-based models to a degree, but are also calibrated or validated using historical data. As such, these models are usually applicable to a range of designs that extrapolate beyond the historical database *to a degree*, but are inappropriate for completely unconventional designs.

Since physics-based models require no calibration other than the validation of the mathematical phenomena being modeled, their infusion into the FLAMES example models is the preferred approach. The use of simplified physics-based models is sufficient to demonstrate the proposed methodology; however, the object-oriented nature of the FLAMES framework supports a variable-fidelity approach to modeling that facilitates replacement with higher fidelity models over time.

Hypothesis 4.1*: Physics-based models, implemented across hierarchical levels, are most effective because they can be mathematically validated.*

C.2 Hypothesis 4.2: k-Factors are an Effective Way to Quantify Technology Impacts

One of the research questions in Section 3.2 identifies the need for a methodology for quantitative technology identification. Since many phenomenological tools "are typically based on regressed historical data, limiting or removing their applicability to exotic concepts or technologies," one useful way in which the impact of technologies can be quantitatively measured is through an application of a *technology k-factor* to the discipline-level metrics of the appropriate contributing analysis in the military simulation hierarchy [293].

k-factors can be seen as scale factors on these discipline-level metrics, either increasing or decreasing the value from a baseline. An example of such metrics for an aircraft includes the lift-to-drag ratio, TSFC, drag coefficient, and empty weight. Infusing a new, lightweight materials technology could be tracked as a decrease in k-weight. Using a lighter empty weight as an input to a range analysis program would result in an increase in aircraft range due to the ability to carry more fuel for the same gross takeoff weight. Similarly, decreases in drag coefficient could be correlated to increased acceleration or turn performance in combat maneuvers. An example of the use of technology k-factors for an aircraft sizing routine is shown in Figure 169. In this example, a simple sizing routine based on the Breguet range equation was used to evaluate range and gross takeoff weight while the cruise speed was calculated from a force balance for straight and level flight using a generic equation for engine lapse ratio from reference [278]. The prediction profiler shown in Figure 169 depicts the partial derivatives of aircraft range, cruise Mach number, and aircraft weight as a function of lower-level design parameters and four technology k-factors.

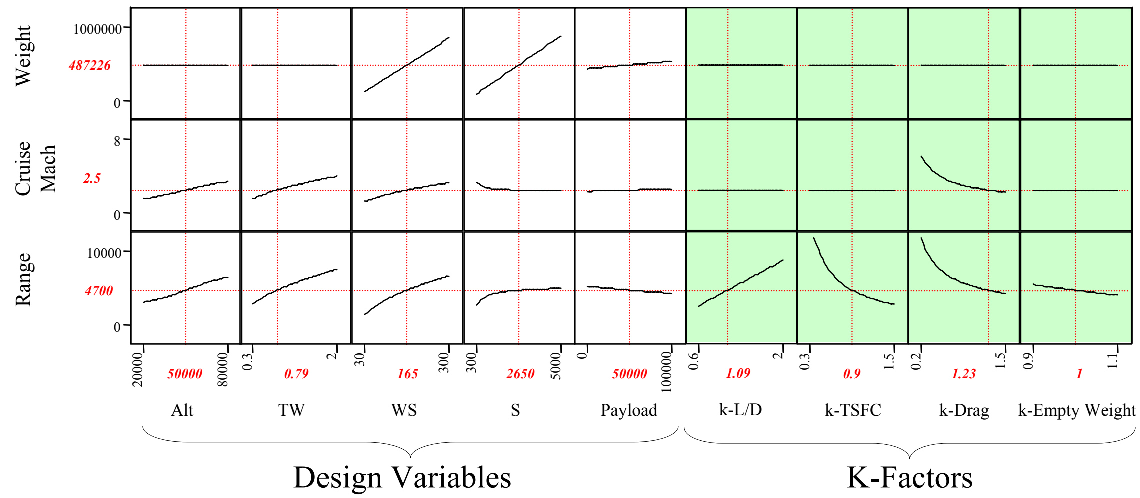


Figure 169: Using k-Factors in Aircraft Design.

This **quantitative** method of technology assessment has been demonstrated for several integrated engine/airframe combinations and also across a military simulation hierarchy [240, 79, 378]. Mavris, Ender and Biltgen extended this hierarchical approach to map k-factors using a top-down approach to identification of system-level and capability-level attributes [280].

This k-factor approach can be extended to the capability-level by tracking the performance of increased range, higher speed, and lower weight solutions against capability-level MoEs. Using a hierarchy of linked surrogate models with k-factors as inputs, technologies at the system and subsystem level can be traced to high-level capabilities. Finally, a probabilistic approach to technology modeling used by Kirby can be implemented using the linked surrogate model approach [240]. Distributions of k-factors can be applied at the system and subsystem levels. These distributions result in “constellations” or “islands” of capability at the top level resembling the Joint Probability Distribution functions shown in Figure 193 in Section C.11.

The k-factor technique has been proven to work well with surrogate models and is used in conjunction with the methodology described in Section B.3 to optimize the selection of tactics as technology k-factors are changed to represent technology infusion, technology refresh, and spiral development approaches.

Hypothesis 4.2: *The concept of k-factors has emerged as a useful method for mapping technology impacts to surrogate model inputs.*

C.3 Hypothesis 4.3: A Hierarchical Tradeoff Environment is Needed for Technology Analysis

Baker formulated a tradeoff environment that uses surrogate models to assess the simultaneous impact of requirements, vehicle characteristics, and technologies for the design of aircraft [50]. This technique is called the Unified Tradeoff Environment (UTE) and was further extended by Soban, who noted that the necessity to analyze a vehicle in the correct context "evaluated as a system fulfilling its intended function" requires that the top level measures of effectiveness are functions of the measures of performance at the mission and vehicle level. She thus formulated a multi-level UTE, shown in Figure 170 to link surrogate models at various hierarchical levels, represented graphically by a prediction profiler. The metrics along the y-axis of the prediction profiler become the inputs along the x-axis at the next highest level. When analyzing a multi-domain problem, it is possible that continuous variables may not be able to track multiple systems. An aircraft has design variables such as wing area and TSFC while a ballistic missile use mass ratio and specific impulse. The potential for discontinuous options further drives the development of the UTE to the formulation shown in Figure 171, which shows how multiple discrete systems may provide measures of performance to the mission and campaign level UTE's. Furthermore, certain types of technologies at the subsystem level may be applicable to multiple systems. This underscores the complexity of the system-of-systems problem: *everything is related to everything.* A screening approach for systems-of-systems is needed that identifies the subset of parameters across all systems and technologies to allow decomposition of the problem to a manageable size.

Hypothesis 4.3: *A hierarchical, integrated tradeoff environment is needed to analyze metrics at multiple levels.*

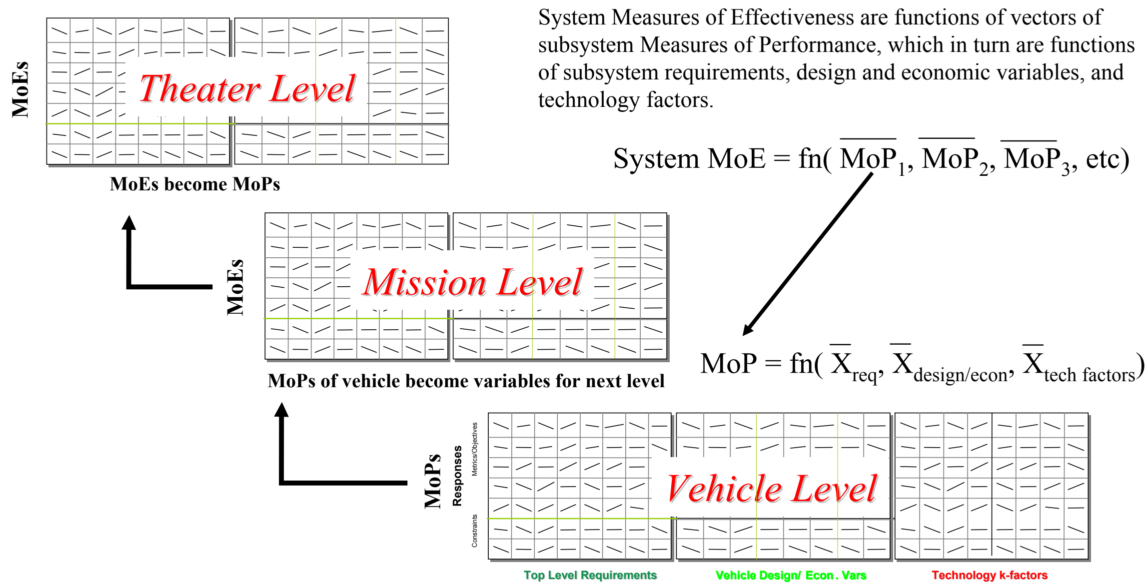


Figure 170: Multi-Level Unified Tradeoff Environment for Analysis of System Effectiveness [378].

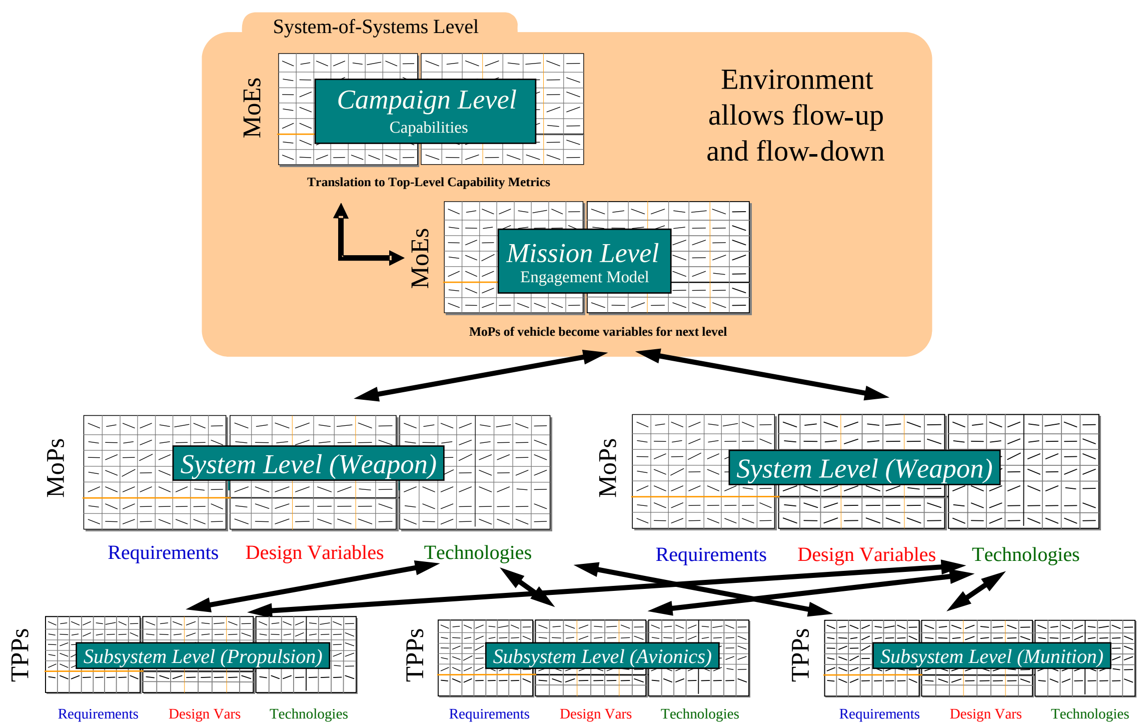


Figure 171: Multi-Level Unified Tradeoff Environment for System-of-System Analysis of Multi-Domain Problems [62].

C.4 Hypothesis 4.4: Bottom-Up and Top-Down Analysis is Used for Capability Assessment

The primary benefit of surrogate modeling approaches is the ability to perform bottom-up or *exploratory design*. Exploratory design, also called parametric design, is when surrogate models are purposefully varied in a deterministic manner to examine the impact of a specific change in design on the overall response. This is identical to running the actual simulation as an analysis tool with the notable exception of decreased run time. The campaign tool mentioned in the previous sections takes approximately 9 minutes to generate a single run. Embedded inside this tool is a missile analysis code that takes approximately 5 minutes to run. Manual transfer of data between the two codes also slows the analysis process. By encapsulating the missile code within the campaign code and generating neural network approximations around them, the user can get reasonably accurate estimates of the model responses in less than a second instead of thirty minutes.

Exploratory design allows a user to try a myriad of designs very quickly simply by changing the input parameters of the response surface equations. This can be done by physically entering all model coefficients or by using the prediction profiler tool shown in Figure 191. As a matter of process, a deterministic exploratory design environment is a **requirement** and enabler for probabilistic design techniques[1]. To be able to do probabilistic design, one first has to master deterministic, parametric design.

This design technique is useful for system-of-systems approaches as inputs for different systems can be varied using a one-variable-at-a-time approach. In this manner, a system architecture is visually optimized and multiple solutions that satisfy the same top-level criteria can be manually located. One of the primary goals of this work is to enable system-of-systems level trade studies to be performed at the system and subsystem level. While exploratory design offers a way to perform these trades using a brute force approach, the next section addresses the topic of *inverse design*, which uses probabilistic techniques to partially automate the search for an elegant solution to the same problem.

[1]This is not true if the run time of the suite of analysis tools is already shorter than one second. To date, no reasonably accurate engineering application for aircraft design that meets this constraint has been demonstrated.

Probabilistics coupled with surrogate models are the two enabling techniques for *inverse design*. Similar to the popular Cost as an Independent Variable (CAIV) approach advocated in DoD 5000.2-R, inverse design treats *any variable as an independent variable*. In capability-based design, the approach allows "design for capabilities."

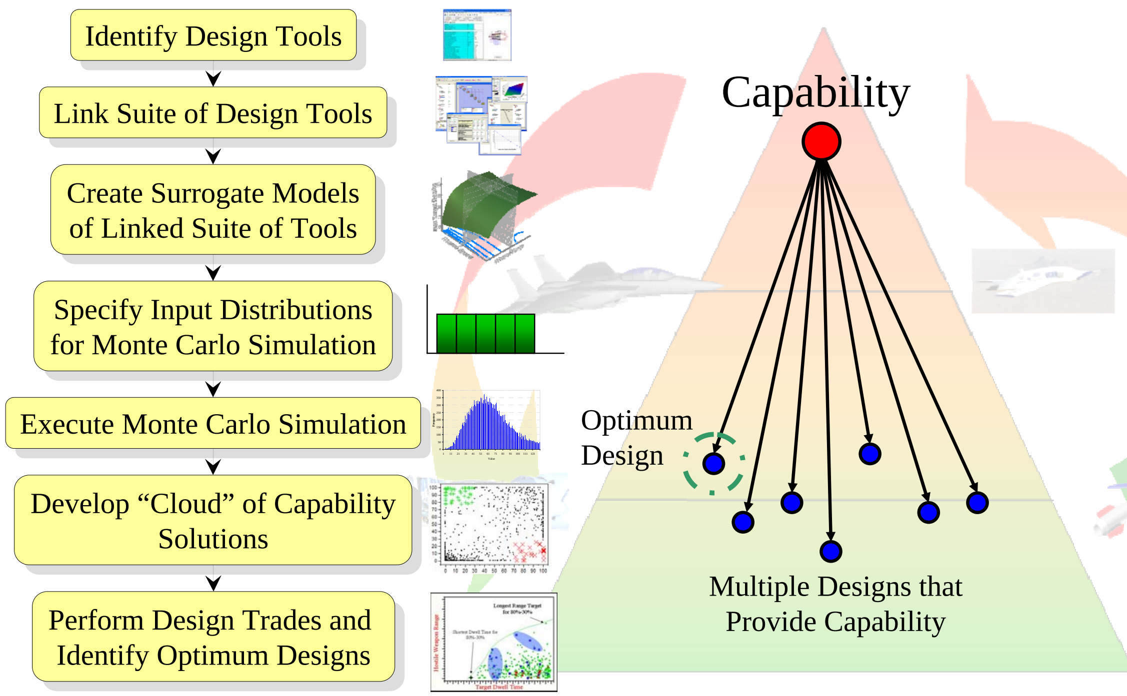


Figure 172: The Steps of the Inverse Design Process.

The inverse design approach is illustrated in Figure 172. Once a suite of design tools has been identified and linked, surrogate models are created for the suite of tools. The purpose of this step is to speed up the process to the point where running cases is trivial as a large number of cases are needed to do trade studies for a system-of-systems process[2]. Next, a Monte Carlo simulation is used to execute the suite of tools over a range of input variables at various hierarchical levels. **Inverse design is actually accomplished by doing forward design many times using MCS**. The results of the Monte Carlo simulation are "clouds" of capability solutions at various hierarchical levels. Analyzing these clouds using

[2]Knowledge gathering via simulation is a function of computing power and available time. Surrogate models decrease the amount of time required for finite computing power. As computers advance, designers can choose to get the same information in less time or more information in the same amount of time. Historical trends indicate that design organizations prefer the latter in most cases.

a multivariate profiler allows rapid design trades and identification of regions of interest.

Using this technique, it is possible to highlight a desired capability in relation to system-of-systems level metrics and identify potential designs at the system and subsystem levels (See Figure 172). The relationship between forward design and inverse design is shown in Figure 9. Although systems engineering is a top-down process, many engineers "think" in bottom-up terms because that is what they do: make purposeful changes in the things they can control and examine the outputs of the process. The inverse design process, aided by functional decomposition, the DoDAF views, and the matrix of alternatives technique, is aimed at identifying physical systems that provide the required capabilities. A bottom-up reconciliation process utilizing a hierarchical modeling and simulation environment is still needed to assess *how* the physical systems perform with respect to capabilities. As new technologies develop and capabilities evolve, the process shown in Figure 9 continually reconciles the differences between capabilities desired by the warfighter and those provided by system designers.

The inverse design process is best viewed through a multivariate analysis matrix that relates each variable to every other variable. This graphical technique, shown in Figure 173 is useful for analyzing any variable as an independent variable. In a linked, graphical analysis environment such as the JMP® scatterplot matrix, highlighting desired thresholds for capability-level MoEs in the top left corner simultaneously highlights those points in all other dimensions. This technique is useful for identifying trends and groupings. Multidimensional behaviors can be viewed by human operators using a combination of colors, symbols, density plots, and constraint lines.

In the example shown in 173, capability-level MoEs are a function of both enemy attributes and friendly attributes (the top-level "circuit" is identified by a dotted black line). The attributes of the friendly system are in turn specified by system and subsystem-level surrogate models for an aircraft, weapon, and weapon engine. Using a multivariate plot in JMP®, it is possible to highlight points (representing system solutions) that meet certain thresholds with respect to any variable and identify where these points fall in every other dimension.

A simplified form of the multivariate profiler for the military simulation developed by Ender is shown in Figure 174 [289]. This example uses a notional Long Range Strike asset firing powered missiles in a battlespace occupied by SAMs and TCTs. The system-of-system level MoEs representing capability metrics are shown in the upper left hand corner while system and subsystem design parameters are shown in the lower right corner. Several conclusions can be drawn by visually examining the dispersion of points. The first plot shows targets killed versus platforms lost. The desirable region for this plot is minimum platforms lost and maximum targets killed, identified by the green highlighted region. The red region indicates the least desirable system solutions in regards to these two metrics. The second plot shows enemy capability in terms of the range and speed of hostile SAMs. In this plot it is evident that the green points prefer the lower half of the plot and the red points have a greater population in the upper half: our systems perform better against SAMs with low shoot-back range. Next, friendly capability in terms of weapon range and weapon speed is examined. In this plot, the green points favor long range, high speed weapons and the red points represent "low capability" friendly weapon solutions. The lower right corner plot shows cruise Mach number versus diameter of the weapon. The results are somewhat inconclusive at first glance, although statistical tests on the red and green regions provide salient information about the region as a whole. Additionally, the conclusions can be reached more definitively by narrowing the design space through successive eliminations of designs.

Since only the green region is desired, the black and red points in Figure 174 can be hidden to clarify the decision making process. These points represent solutions that do not meet a threshold of at least 80% targets killed and no more than 30% platforms lost. A zoomed in version that examines blue capability against red capability for the green points identified is shown in Figure 175. In the left side of the figure, red points representing short range, subsonic weapons can he highlighted and colored red. Blue points representing long range hypersonic missiles can be colored blue. The right side of Figure 175 *identifies what types of targets in terms of enemy capability blue forces can prosecute with different levels of friendly capability.*

The concept of the ***Pareto Frontier*** is useful to explain optimality in two simultaneous

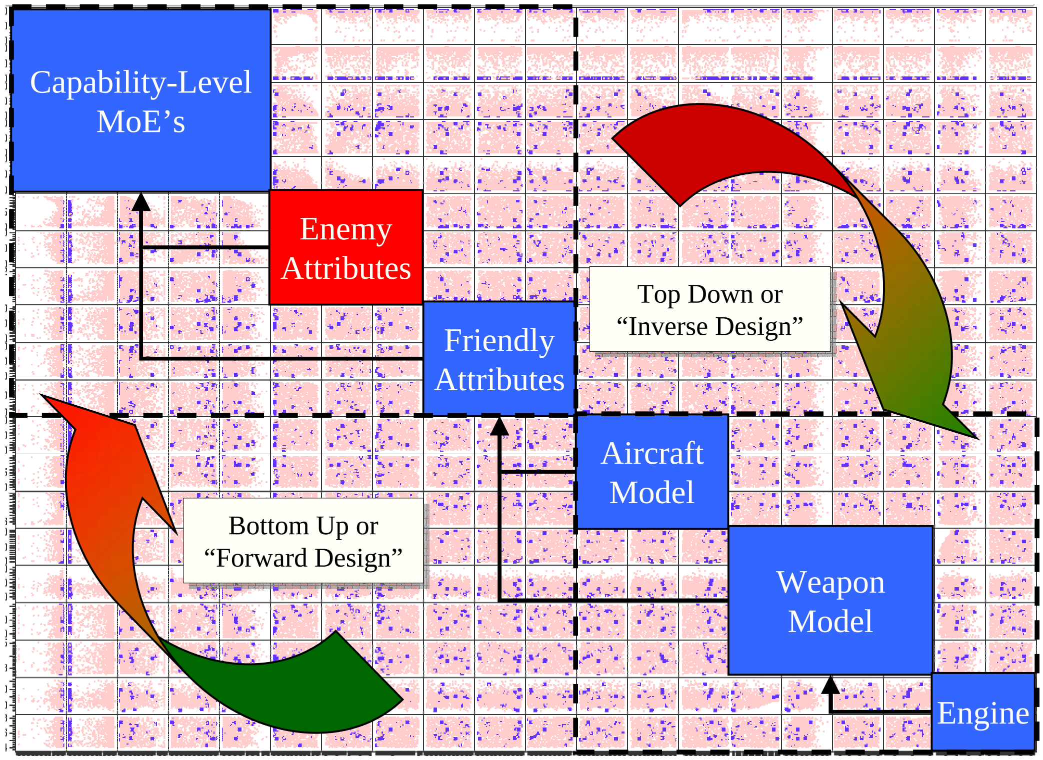


Figure 173: Multivariate Plot for a System-of-Systems Problem.

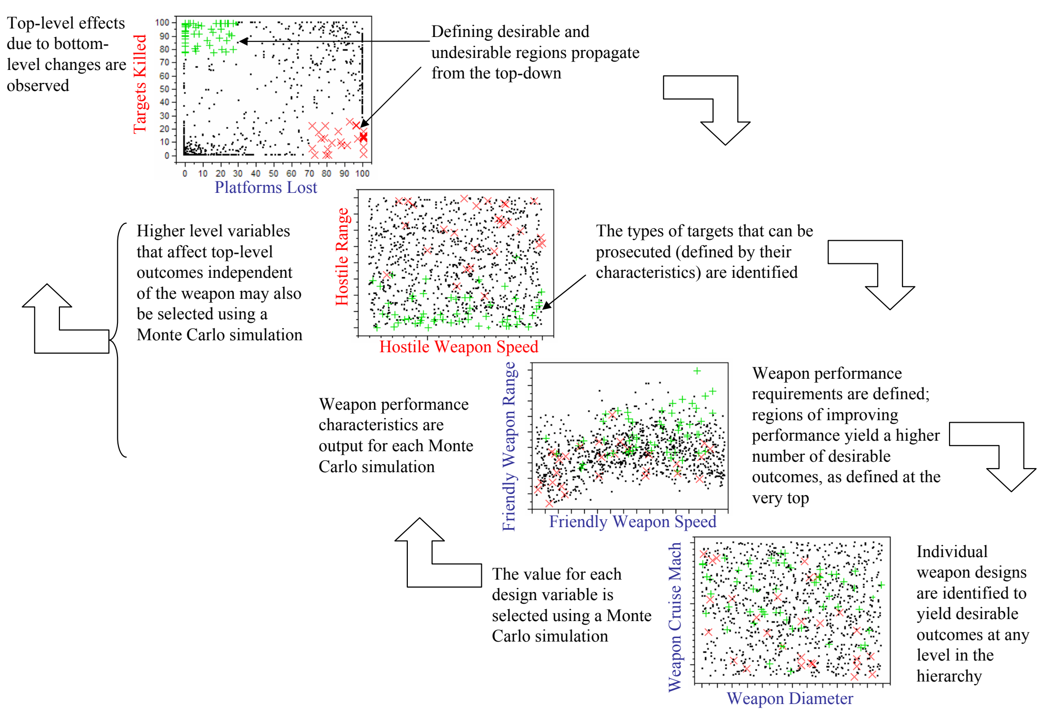


Figure 174: Using Inverse Design to Discover Capability Solutions (Courtesy of Ender [289]).

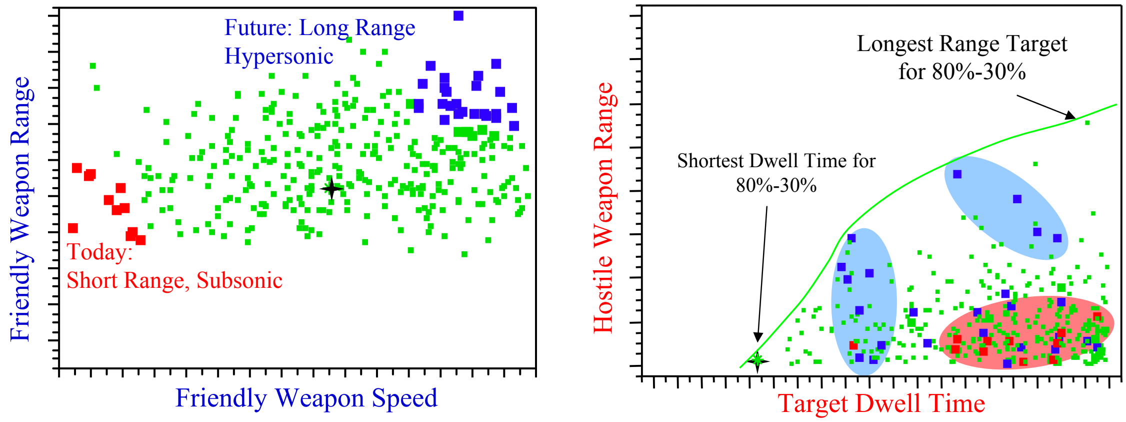


Figure 175: Using Inverse Design to Discover Capability Solutions at the System Level [289].

dimensions. In a two dimensional space, a point that is *Pareto Optimal* is one that has maximal utility in both dimensions with respect to all other points in those dimensions. Pareto optimal points represent an optimal design depending on how the two dimensions are weighted. A Pareto frontier is therefore the locus of all Pareto optimal points in those two dimensions as depicted by the red points in Figure 176.

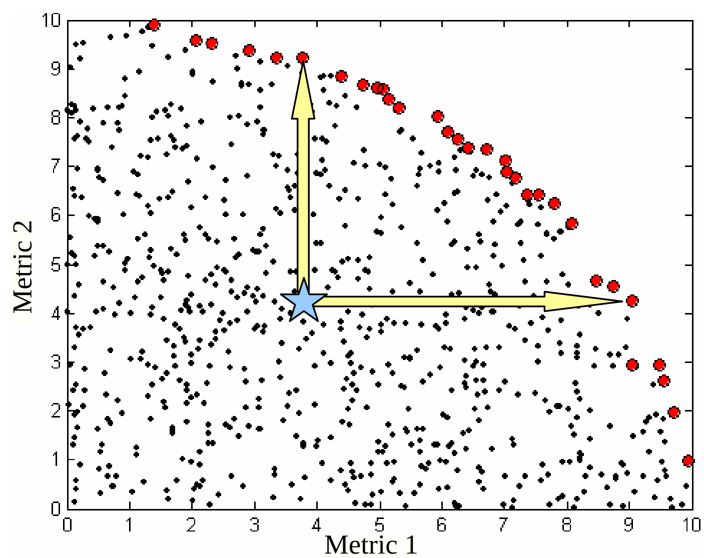


Figure 176: Illustration of a Two Dimensional Pareto Frontier [270].

In Figure 175, the curved green line represents the Pareto frontier for enemy capability: blue cannot prosecute targets with greater range or shorter dwell time than this frontier without losing more than 30% of platforms or killing less than 80% of targets. As the right side of the figure shows, the blue points tend to be closer to the Pareto frontier than the red points. Short-range subsonic solutions cannot prosecute targets with short dwell time *or* long shoot-back range. This plot shows that developing long range hypersonic missiles has a beneficial impact on blue's overall capability. The multivariate profiler can be further used to examine specific technologies that provide this capability, evaluate the development or production cost of candidate solutions, or impose additional design constraints that could not be modeled using physics-based tools and military simulation codes.

As noted by Daskilewicz and German, the identification of the Pareto frontier is less obvious in multiple dimensions. In Figure 177, a three-dimensional Pareto surface that represents optimal points is highlighted with red symbols.

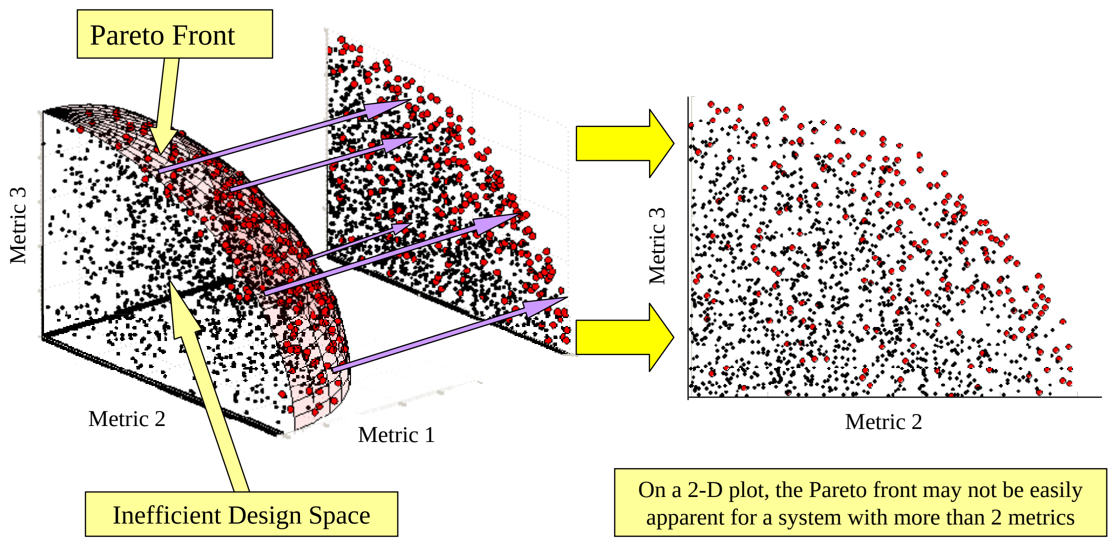


Figure 177: Illustration of a Three Dimensional Pareto Frontier [270].

When this surface is viewed in two dimensions, Pareto optimal points appear dominated in one or more of the dimensions although they are optimal in a third. The multivariate inverse design technique seeks to identify multidimensional optimality and trace desired capability thresholds to individual design solutions.

Inverse design is a powerful technique that enables decision making for systems-of-systems. Because complex interdependencies in the system-of-systems are not known *a* *priori*, a large number of cases must be run to provide good coverage in the multivariate plot and discover system solutions that satisfy MoEs at the capability level. This is accomplished using a Monte Carlo simulation over the range of inputs (forward design) and viewed from a capability-standpoint (inverse design). Because the number of cases needed is so large, surrogate models allow evaluation of these cases in a reasonable amount of time. Finally, software and visualization techniques are needed to assimilate the large amounts of data generated in a system-of-systems design problem so that reasoned and defensible decisions can be made regarding capabilities, systems, and technologies.

Hypothesis 4.4*: The inverse design technique is useful for setting targets at the top level and tracing these targets to measurable system attributes and technology performance metrics.*

C.5 Hypothesis 4.6: A Matrix of Alternatives and the Systems Modeling Language are Useful in Reducing the Scale of the Problem

Part of the complexity in systems-of-systems problems is caused by their large scale, which breeds many cross-system interactions. In this section, techniques for mapping the dimensionality of the problem and separating those SoS elements that contribute significantly to the MoEs of interest are described. The first technique, the matrix of alternatives, is primarily used for the design of the conceptual model while the SysML technique can be used to diagram the conceptual model and implement the computer model.

C.5.1 Matrix of Alternatives

Morphological Analysis (MA), a revised version of a technology forecasting approach postulated by Mendeleev in 1869, was developed by Fritz Zwicky, a Swiss-American astrophysicist and aerospace scientist in 1966 [415]. MA is "a method for investigating the totality of relationships contained in multi-dimensional, usually non-quantifiable problem complexes" [350]. This non-quantifiable analysis often relies on judgement more than analytical results. From the Greek *morphe*, meaning shape or form, the general definition of morphology is "the study of form or pattern" [350]. Zwicky generalized the technique from scientific fields such as anatomy, botany, geology, and biology.

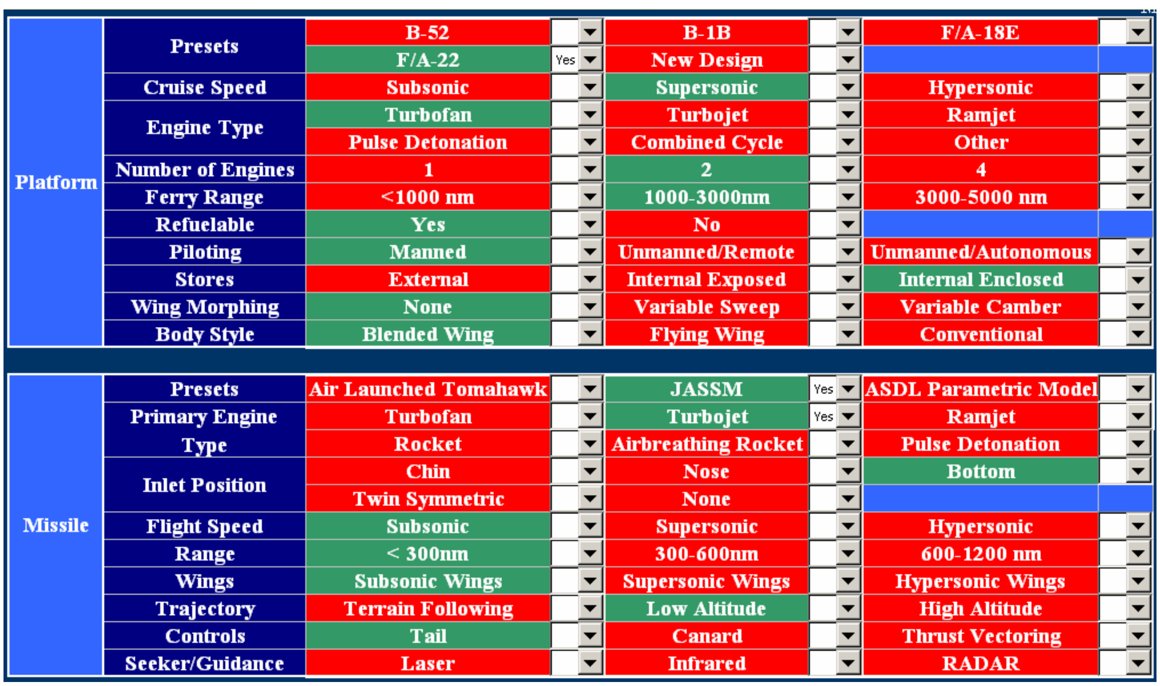
MA is performed in the systems engineering process using a morphological matrix. Also called a matrix of alternatives, the technique is a decomposition of the important parameters for a given problem and a listing of potential solutions for each dimension often by decomposing the solution space into a series of rows and columns. An example of a matrix of alternatives for selecting a pair of pants is shown in Figure 178. In this example, the various factors that describe a pair of pants are categorized and listed down the left side while the options for each factor are enumerated across the columns. Although this matrix only has fourteen rows and a handful of options per row, there are over 26 million combinations that can be synthesized from the options provided. By selecting a single design option in each row as shown in Figure 178, this combinatorial space can be narrowed to a single option.

Physical Constraints	Waist	32	34	36	38	40	
	Inseam	30	32	34	36	Other	
Comfort	Composition	Cotton	Polyester	Rayon	Other		
	Fit	Classic	Relaxed				
	Elastic Stretch	Yes	No	Hidden Waistband			
Style	Color	Navy	Black	Khaki	Mocha	Other	
	Cuff	None	Cuffed				
	Pleated	Pleated	Double Pleat	Flat Front			
	Flare	Yes	No				
Maintainability	Wrinkle Resistance	Wrinkle Resist	No Iron	Wrinkle Free	None		
	Stain Resistance	Yes	No				
	Machine Washable	Yes	No				
	Fade Resistant	Yes	No				
Other	Brand	Alfani	Dockers	Tommy	Club Room	Timberland	Other

Figure 178: Matrix of Alternatives for Selection of a Pair of Pants [144]

Recent research led by Engler has developed the concept of the Interactive Reconfigurable Matrix of Alternatives (IRMA) [144]. IRMA combines design choices with a compatibility matrix to calculate the number of feasible design options after each downselect. Using multi-attribute decision making, filters can be applied at each functional row of the matrix to justify why a given concept is selected. Filters can be applied on the basis of cost, schedule, technology readiness level, policy concerns, or other factors. The IRMA also provides insight into how long it takes to analyze all the combinations produced by the matrix of alternatives. Systems-of-systems design requires the use of morphological matrices to focus the modeling effort on the necessary elements. The engineering time required to develop a myriad of models and the computational time to execute them even for the simplest parametric study is on the order of decades. The IRMA facilitates reasoned, defensible downselects of the product attributes to a concept space that can be adequately modeled within a given time.

An IRMA for a Long Range Strike aircraft that uses powered weapons is shown in Figure 179. This IRMA categorizes the LRS system into platform and missile options, and enumerates several design choices for each factor. Using the drop-down options, the user can select whether or not a particular option is to be considered and calculate how many architecture options remain.



Platform	Presets	B-52		B-1B		F/A-18E	
		F/A-22	Yes	New Design			
	Cruise Speed	Subsonic		Supersonic		Hypersonic	
	Engine Type	Turbofan		Turbojet		Ramjet	
		Pulse Detonation		Combined Cycle		Other	
	Number of Engines	1		2		4	
	Ferry Range	<1000 nm		1000-3000nm		3000-5000 nm	
	Refuelable	Yes		No			
	Piloting	Manned		Unmanned/Remote		Unmanned/Autonomous	
	Stores	External		Internal Exposed		Internal Enclosed	
	Wing Morphing	None		Variable Sweep		Variable Camber	
	Body Style	Blended Wing		Flying Wing		Conventional	
Missile	Presets	Air Launched Tomahawk		JASSM	Yes	ASDL Parametric Model	
	Primary Engine Type	Turbofan		Turbojet	Yes	Ramjet	
		Rocket		Airbreathing Rocket		Pulse Detonation	
	Inlet Position	Chin		Nose		Bottom	
		Twin Symmetric		None			
	Flight Speed	Subsonic		Supersonic		Hypersonic	
	Range	< 300nm		300-600nm		600-1200 nm	
	Wings	Subsonic Wings		Supersonic Wings		Hypersonic Wings	
	Trajectory	Terrain Following		Low Altitude		High Altitude	
	Controls	Tail		Canard		Thrust Vectoring	
	Seeker/Guidance	Laser		Infrared		RADAR	

Figure 179: Interactive Reconfigurable Matrix of Alternatives (IRMA) for a Long Range Strike System Using Air Launched, Powered Weapons [144, 289]

While an IRMA for an LRS system architecture can comprise air, land, space, and near-space assets operating at multiple theaters to perform multiple capabilities, simplified versions of the matrix of alternatives for friendly and hostile assets are used in the proof of concept demonstration in Figures 48 and 49.

Finally, in addition to its use for concept selection, the matrix of alternatives is also used throughout this work to identify the methodology options at each phase of the synthesis process and highlight downselections as they are made. Examples include Figures 20 and 22.

The Systems Modeling Language, or SysML is a subset of the Object Management Group's UML 2.0 specification [388]. Its parent, the Unified Modeling Language (UML) is a modeling and specification language used by software engineers, but can also be used to model hardware, business processes, organizational structure, and systems engineering [16].

SysML is an extension of UML 2.0 for systems engineering and has support from the International Council on Systems Engineering (INCOSE) and the Object Management Group (OMG). SysML grew out of the shortcomings of the UML 1.1 standard including the lack of the deployment diagram to address hierarchical system architectures, the inability of the object sequence diagram to link to sub-sequences or extended Use Cases, the absence of linkages to requirements, and the inability to model parametric equations [194]. The UML 2.0 standard improved this situation somewhat, but only allowed a single level of hierarchy and did not address the lack of requirements and parametric equations. The goal of the SysML "customization" of UML 2.0 is:

> "*to provide a standard modelling language for systems engineering to analyse, specify, design, and verify complex systems, intended to enhance systems quality, improve the ability to exchange systems engineering information amongst tools, and help bridge the semantic gap between systems, software, and other engineering disciplines*" [387].

SysML uses some of the extensions of UML 2.0 and also extends the language with special classes for systems engineers as shown in Figure 180. The purpose of these specification languages is to map out complex engineering projects, thus eliminating duplication of effort and ensuring that the appropriate interactions are tracked throughout the product life cycle. The UML, adopted in 1997, is used to create meta-models, which in software engineering refer to high-level abstractions of code. Another way of describing software modeling is *designing software without writing code.* This is useful for large, distributed software engineering projects where geographically distributed programmers write pieces of code that must be integrated into the entire enterprise level product. The parallel to system-of-systems engineering is direct.

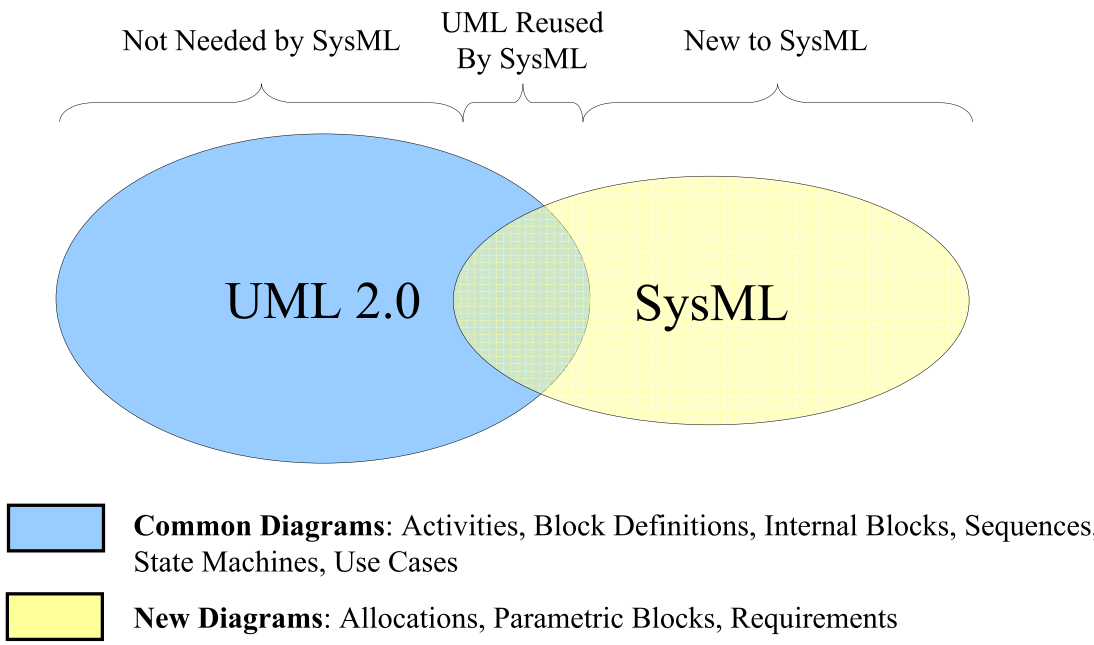


Figure 180: Overlap between UML 2.0 and SysML Standards. (Adapted from [12])

As illustrated in Figure 180, the objective of the SysML group was to use as much UML 2.0 as possible and avoid excessive creation of new types of diagrams. In fact, the requirements diagram and parametric equations diagram are variations on existing UML 2.0 diagrams [194]. Requirements diagrams also provide a means to interface with textual requirements that may be held in a resource repository. This is an important element of *requirements analysis* and *requirements management.* The parametric equation diagram allows equations that relate parameters in the model to be included in the SysML depiction. When complex physics-based analysis tools are used at the system level, these diagrams tend to be redundant. The parametric equation diagram is useful if the SysML is used as the actual programming language for system design.

Aside from different diagrams and nomenclature, the use of the two standards is the same. According to Fowler, there are three primary ways of using the UML: sketch, blueprint, and programming language [155]. Sketches are used to roughly diagram approaches and alternatives early in the design process and are analogous to conceptual and preliminary design. Blueprinting, on the other hand, involves the detailed creation of the entire structure of the code. This is similar to the detailed design phase of aerospace engineering where specification drawings are prepared for manufacturing. Lastly, while the UML can be used as a programming language, it is often not efficient to do so: the sketches and blueprints are independent of programming language. In this research, the sketch and blueprint approach is used to lay out the structure of the LRS problem for implementation in FLAMES.

UML and SysML are *standards*. They define a method for creating diagrams that describe a product or process. While the rules defined by the UML and SysML standards can be implemented using a white board or pencil and paper, a number of software tools including Rational® Rose®[3], Telelogic® System Architect[4], and ARTiSAN Studio[5] have been developed. These tools provide a user interface for the creation of UML diagrams. As indicated by the footnotes, there is much change and consolidation in the UML tool industry to keep up with diverse customer needs and the growing desire for major institutions to align their enterprise-wide software products. All three tools support the DoDAF architecture views, and both Rational Rose and ARTiSAN Studio have been used to diagram the LRS architecture. As of October 2005, "ARTiSAN is the only vendor that offers and implementation of the current specification of SysML" [4]. As an example, the DoDAF OV-2 view for mission planning shown in ARTiSAN Studio in Figure 181.

To demonstrate the use of SysML for blueprinting code modules, an activity diagram for a SEAD aircraft is shown in Figure 182. Activity diagrams "are a technique to describe the procedural logic, business process, and work flow" [155]. They are similar to flowcharts and functional flow block diagrams except that they allow parallel behavior which in software

[3]IBM acquired Rational Software on December 6, 2002 and has integrated Rational tools with its enterprise software solutions [11]. The Rational product catalog has been recently extended to include Software Architect, Software Modeler, and System Developer.

[4]Telelogic acquired Popkin Software on April 18, 2005 [32]. Telelogic also makes the DOORS requirements management product.

[5]Formerly Real Time Studio

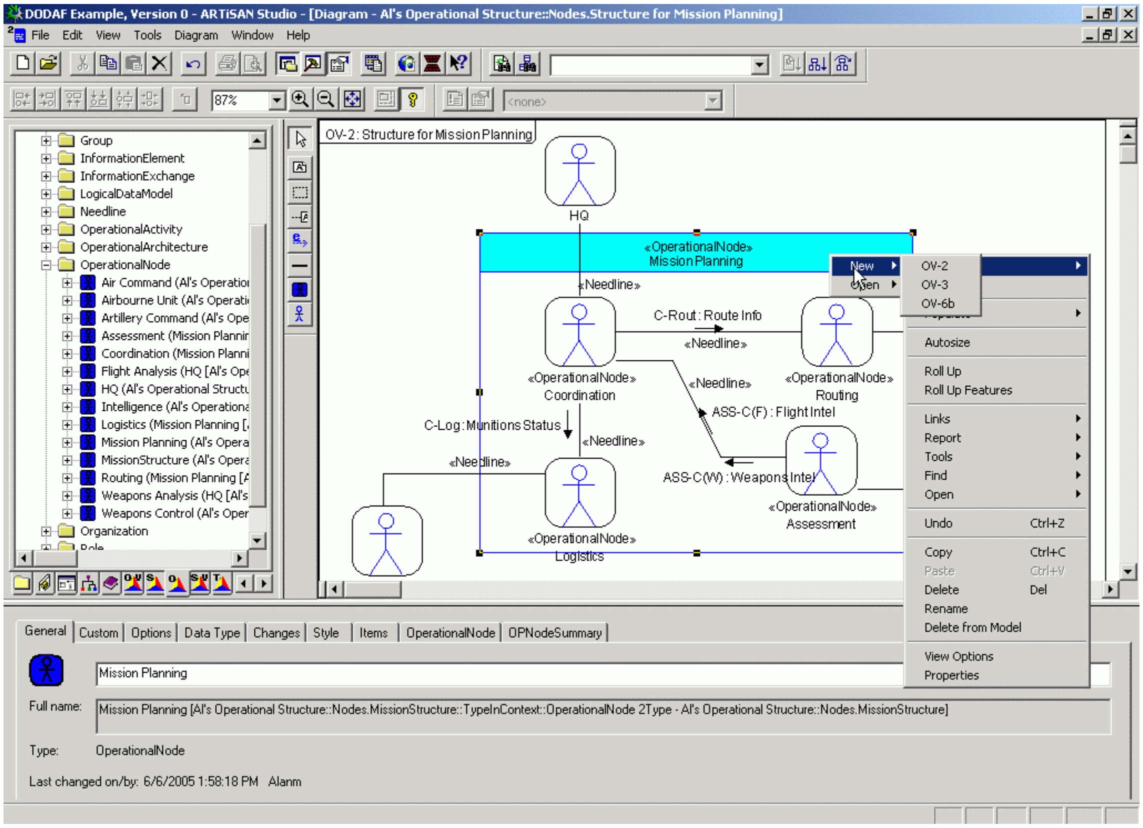


Figure 181: Using ARTiSAN Studio to Model DoDAF OV-2 View [349].

engineering indicates that multi-threading or parallel processing can be used to handle more than one activity simultaneously. In military simulation, the activity diagram can be used to describe the potential actions a vehicle can perform which is useful in mapping the cognitive behaviors that must be developed to allow the asset to perform the identified actions. Activity diagrams are also similar to Petri Nets [471]. An activity diagram for a fighter aircraft flying strike or SEAD is shown in Figure 182.

Sequence diagrams "describe how a group of objects collaborate in some behavior" [155]. A sequence diagram is subtype of interaction diagrams along with the communication diagram, interaction overview diagram, and timing diagram. Sequence diagrams are similar to functional architectures and functional flow block diagrams [130]. These are useful in planning how systems interact in a system-of-systems architecture and can be used to model network centric operations.

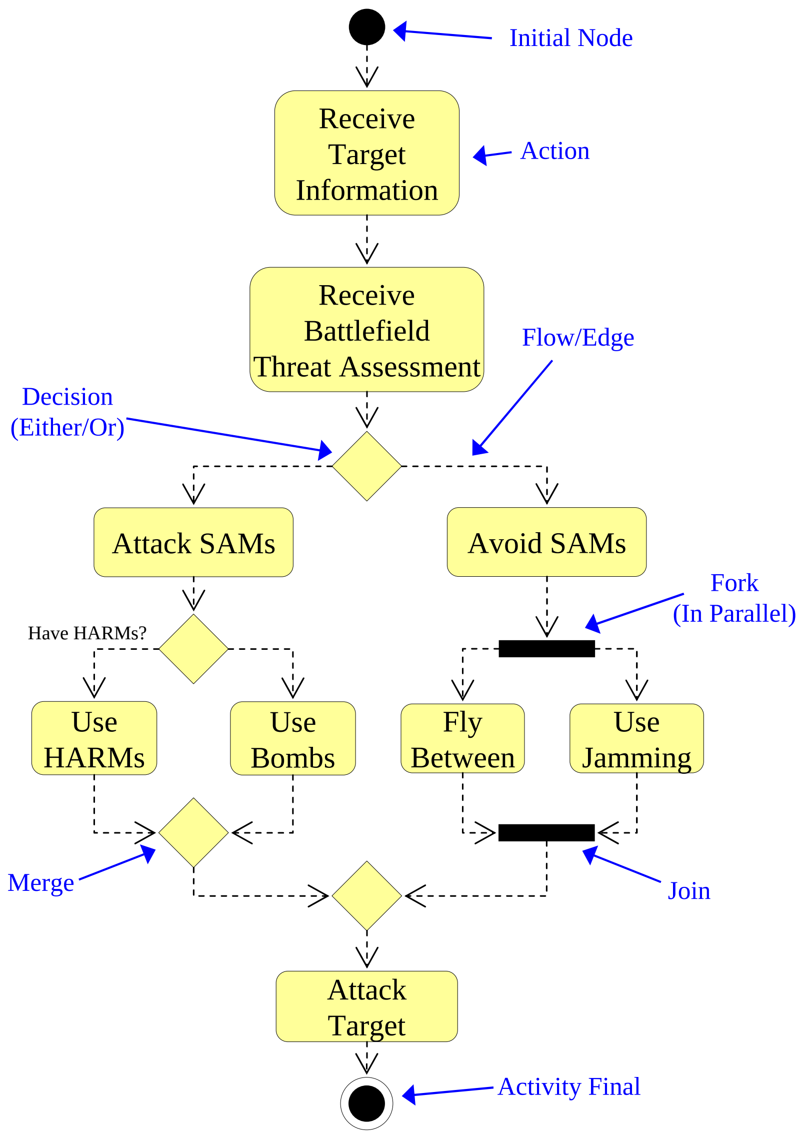


Figure 182: Example Activity Diagram for a Fighter Aircraft.

State machine diagrams (or state diagram) are used to describe the behavior of a system by describing certain "states" and the methods used to traverse from state to state. These diagrams are another way of looking at the same type of information in a sequence diagram. Instead of focusing on the actions (verbs) like the activity diagram, a state diagrams focus on the states (nouns). Example states for a fighter aircraft include waiting on the runway, taking off, weapons locked onto target, bomb bay open, under attack from the ground, under attack from the air, or dead. Various methods define how the aircraft moves from state to state, for example, if the "intercept blue fighter with red missile" method is used successfully, the state of the blue fighter changes from "under attack from the air" to "dead." State machine diagrams are used to outline the necessary physical and cognitive elements needed to create an object-oriented constructive simulation for the Long Range Strike system.

The last diagram that is critical in this research is the Use Case Diagram. Use cases are "the specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system" [388]. SysML does not provide a means for *writing* use cases, rather, the use case diagram allows the designer to amalgamate all the created use cases together. Given a number of use cases for the systems in an architecture, the use case diagram can help systems engineers plan scenarios and missions that involve the elements of the architecture.

Why are these standards considered "advanced design methods?" While software modeling has been popular since the 1980's and ubiquitous since the mid-1990's, detailed conceptual modeling for systems-of-systems has not seen *widespread* acceptance until the adoption of the DoD Architecture Framework standard in August 2003[6]. In fact, while there are a number of books written about UML for software modeling, as of late 2005, SysML-related literature is limited to white papers and Powerpoint presentations. A structured method for examining the complex linkages in an architecture is important because integrating systems acquired with a service-centric focus into joint operations has shown that after-the-fact integration and interoperability is difficult at best. The implementation of UML and SysML in systems engineering enhances understanding of complex interactions between systems in a system-of-systems architecture and enables concepts like network centric operations and capability-based acquisition.

While many companies have been involved in this type of work, the specific approaches used are usually proprietary. SysML is an advanced design technique because its mastery and mainstream use facilitates the development of a process for system-of-systems engineering that is publicly available.

[6]While the official DoDAF version 1.0 was released on August 30, 2003, precursors to DoDAF were released as the C4ISR framework in 1996 [389]. The United Kingdom Ministry of Defense published version 1.0 of the Ministry of Defense Architecture Framework (MoDAF) in September 2005.

The implementation of the LRS problem using SysML to diagram the necessary elements of the FLAMES simulation is detailed in Section 5.3.1.7.

Hypothesis 4.6*: A matrix of alternatives is a useful technique for reducing the scale of the problem using a defensible downselection process. The Systems Modeling Language is useful for diagramming code and programming modules.*

C.6 Hypothesis 4.7: SWARMing, Brainstorming, and Functional Decomposition are Useful Techniques for Determining the Necessary Elements of a System Architecture

The System Wide Assessment and Research Method, or SWARM, is a technique that is fundamental to the understanding of a problem. SWARMing occurs as the first step in the design process, and is essentially a massive knowledge-gathering technique. As part of a comprehensive problem definition method, SWARMing includes literature searches of a variety of sources, identification of new methods that must be developed in order to analyze and assess the system, and the selection of tools relevant for the design. This initial knowledge-gathering serves as a rapid training phase for project participants. During SWARMing, team members "learn the lingo" of the design community and build a foundation on the relevant systems so they can have relevant and meaningful discussions with the customer. SWARMing builds knowledge and confidence, especially when designing a system outside the realm of expertise of the designer.

The necessary elements for the system-of-system implementation of the Long Range Strike Capability is determined through SWARMing. After a downselection process has been performed, detailed model construction in the FLAMES tool is aided by the SWARM technique, as myriad data on military systems must be collected from a variety of sources to accurately model the aspects of the Long Range Strike system-of-systems. The primary difficulty in system-of-systems engineering is the large scale of the problem. Brainstorming, "a process undertaken by a person to solve a problem by rapidly generating a variety of possible solutions" is "to the morphological method what simulation by Monte Carlo method is to the combinatorial analysis" [22, 234]. While numerous brainstorming techniques abound in the literature, a general tool for this purpose, MatchWare OpenMind, is "a powerful, visual learning tool designed to help you develop and organize ideas. Based on the proven Mind Mapping® theory, it enhances creativity, clarifies thinking and improves memory" [273]. The mind map, an example of which is shown in Figure 47, serves the same general purpose as a large white board and post-it notes: organizing the scope of the project.

The combination of heterogeneous systems to develop the desired effects (caused by the emergent behavior of the system-of-systems) involves complex integration and therefore effective project management skills. The Architecture for Information Systems (ARIS) is a business development tool that allows best practices and processes to be mapped. Files and documents can also be attached to any step of the process, as ARIS functions as a data management system for complex processes [211]. These brainstorming tools help identify which elements *and linkages between them* are required to analyze a desired capability.

Hypothesis 4.7*: SWARMing and brainstorming were identified as techniques that are useful for scoping the problem and defining the appropriate "control volume" to be analyzed using modeling and simulation. A functional decomposition is a systems engineering technique that is useful in relating operational activities to system functions.*

C.7 Hypothesis 4.8: Surrogate Models Enable Rapid High Fidelity Analysis

Unfortunately, a major technical challenge in executing inordinately complex, large-scale constructive simulations is the long run time associated with their execution. According to the National Science Foundation, the way simulations are performed must be revolutionized to "incorporate new discoveries that simplify and enhance multiscale, multidisciplinary simulations" [35]. To avoid "analysis paralysis," a technique is needed to speed up the execution of the simulation to enable large simulations to occur in reasonable timeframes.

Surrogate models[7] are an approximation technique for replacing existing analytical models with a suitable substitute. Surrogate models in the form of response surface equations were first introduced by Box and Wilson in 1951 and developed extensively throughout the

[7]This term has gradually replaced "*metamodel*" in advanced design nomenclature as metamodels are associated with low-fidelity approximations in the software engineering community (see Section C.5.1).

1950's. After several failed attempts in the 1970's and 1980's, the first successful widespread application of surrogate models in the aerospace community was initiated by Tai, Mavris, and Schrage in 1995 [70, 392].

Since the exact relationship between responses and input variables may be difficult to define analytically, a surrogate model is an empirically assumed model that approximates these relationships. These "models of models" can be highly accurate if appropriately created and form the basis of modern advanced design for their wide range of applicability. AFRL Chief Technologist Dr. Thomas Cruse notes that "response surface methods are highly effective over a range of design levels from conceptual to final" [113].

One process by which surrogate models are created is called Response Surface Methodology (RSM). RSM approximates the inherent dependence of functional responses (outputs) to a series of design variables (inputs) using a least-squares regression approach to the determination of unknown model coefficients. The resulting response surface equation (RSE) provides an efficient means to query the design space using a simple equation as opposed to running the full code. The execution of code runs to maximize the effectiveness of the RSM technique with minimal expenditure of computational effort can be realized through application of design of experiments.

Surrogate models provide a number of benefits to advanced design. First and foremost, the replacement of complex design tools with surrogate models has the potential to greatly reduce design cycle time. Although the initial setup of the response surface models can be time consuming, usually a by-product of trying to generate adequate output data at the extremes of the design space using computational tools not designed to do so, the run time with surrogate models is an order of magnitude or more less than that of the actual code. Since RSEs are simple equations, they are not operating system or platform specific. Often, they can be exported into spreadsheets for use as decision making tools for engineers and managers alike.

Addressing one of the technical barriers identified in Section 3.1, surrogate models can handle the dichotomy between large-scale and high detail that is endemic to simulation activities. While a brute force technique would simply run the highest fidelity models available for every asset in the system over every assumption, scenario, and set of doctrine and tactics, a more elegant solution would be to use the ANOVA procedure to assess which of these factors contributes the greatest amount to the variability of the response for the scenario(s) identified. While a full high-fidelity simulation is desired for every radar, missile motor, and turbine blade, realistically something must be given up to explore the design space in a reasonable time frame. Anobject-oriented simulation environment lends itself to a variable-fidelity approach to the analysis of systems-of-systems.

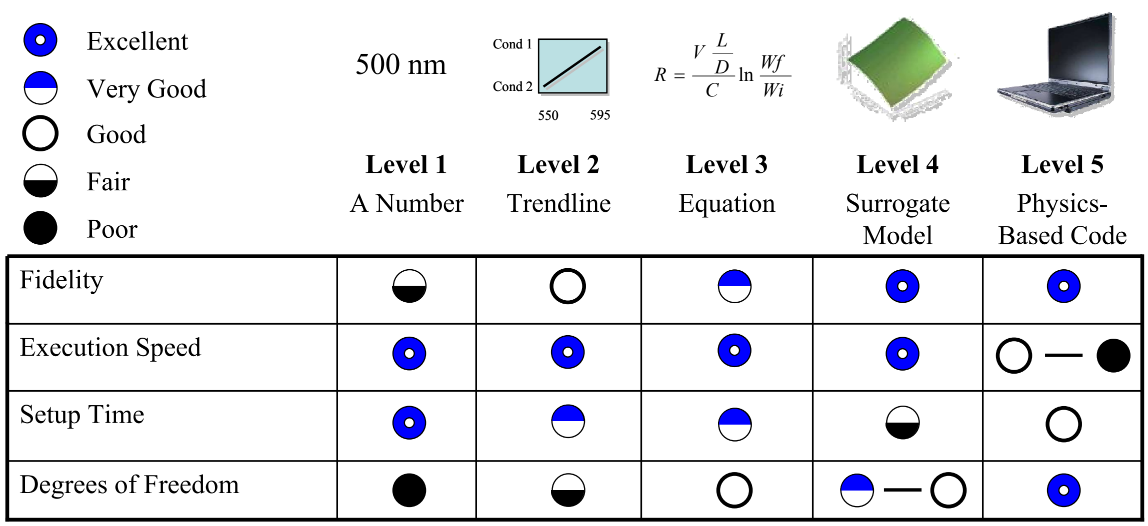


Figure 183: Decomposition of Model Fidelity into Five Manageable Levels.

As shown in Figure 183, this variable fidelity approach can be decomposed into five levels where the lowest level is a simple numerical value and the highest level is a linkage to a physics-based modeling tool. While level five models yield the highest fidelity and degrees of freedom, they suffer in terms of setup time and execution speed. A reasonable compromise is to use high-fidelity surrogate models where appropriate for assets at multiple levels in the military simulation hierarchy. An object-oriented environment is ideal for the variable fidelity approach as multiple levels of fidelity can be encapsulated within a single object. A pseudocode example of this encapsulation is shown below:

```
object AWACS(fidelity_level, vector_of_attributes)
  if (fidelity_level = 1)
     Pfind=0.95
  elseif (fidelity_level = 2)
     if cloudy, Pfind=0.75
     elseif clear, Pfind=0.95
     end
  elseif (fidelity_level = 3)
     Pfind=f(visibility, altitude, power)
  elseif (fidelity_level = 4)
     Calculate RSE using vector_of_attributes (call RSE module)
  else
     Execute physics-based code with state vector
  end
end object
```

A variable fidelity environment is also useful for tracking combinations of qualitative and quantitative information. If it is not possible to develop a physics-based model for a given asset, qualitative values can be used to assess the sensitivity of the capability-level MoEs to the variation of that asset.

Large scale collaborative design is aided by surrogate models. Typically, the primary concern when industry teams collaborate is the protection of intellectual property. Since many behaviors of a computational tool can be encapsulated within a response surface equation, the code itself cannot be reverse engineered. Surrogate models can be created for a specific problem by defining input parameters in a narrow range for the problem of interest. The original code is still needed to examine other problems of the same class. Furthermore, disciplinary experts can create surrogate models using their in-house tools and then export them to other entities as a "currency" of communication in collaborative design activities. This alleviates the need for complex B2B data transfer solutions, collaborative Internet-based infrastructures, or even the physical shipment of man and machine to the collaboration venue.

Surrogate models can also be used to provide intelligence to assets in an agent-based framework. By providing a series of surrogate models that calculate measures of performance based on the state of an asset, intelligent agents can query the provided surrogates to determine their optimum operating conditions. In this manner, these intelligent agents can "know" a lot about their surroundings and their potential to interact with them. This feature is used to provide realistic military campaign simulation without a human-in-the-loop (see Sections 5.5 and 5.6).

Response surface equations have been used in a wide variety of research activities including propulsion systems [244, 358], power systems [320], commercial aircraft [285, 290, 329], unmanned vehicles [295], helicopters [287, 368], tiltrotors [286], missiles [63, 142, 251], surface ships [281, 282], network switches [279], and torpedo design [154, 159]. RSEs have also found use in the design of systems-of-systems including the U.S. air transportation system [127, 161, 258], military aircraft survivability [377, 378], air defense weaponry [141], and Long Range Strike aircraft [289]. This list is certainly non-inclusive: response surface methodology has been the foundation of advanced design research at ASDL since 1995 [392].

Hypothesis 4.8*: Surrogate modeling is a proven approach that balances speed and accuracy to enable rapid design space exploration.*

C.8 Hypothesis 4.9: Neural Networks Enable Modeling of Discontinuities and Nonlinearities

Two types of surrogate models, polynomials and neural networks are summarized below. As noted by Daberkow, neural networks provide advantages for very complex non-linear behavior or large numbers of independent variables when the traditional polynomial approach breaks down [116]. Neural networks are of great interest for the system-of-systems design problem for which this difficulty is notably present.

C.8.0.1 Polynomial Response Surface Equations

A large scope of literature is available on the use of polynomial response surface equations for design (see Section C.7 and this work does not delve deeply into the theory behind them. As an overview, the second-order polynomial equation used is based on a second-order Taylor series expansion:

$$R = b_o + \sum_{i=1}^{k} b_i x_i + \sum_{i=1}^{k} b_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} b_{ij} x_i x_j + \varepsilon \tag{25}$$

Where: y is the approximated response

x_i are the design variables

b_0 is the intercept

b_i are regression coefficients for main effects

b_{ii} are coefficients for quadratic effects

b_{ij} are coefficients for interactions

ε is the approximation error

The coefficients, b, are found through a least-squares regression of data resultant from an intelligently constructed design of experiments. Since the coefficients are not a function of the design variables themselves, the regression is termed linear. The goodness of fit for a polynomial RSE can be verified using the five step process outlined by Barros and Kirby [55]. Typically, face-centered central composite DOEs have been demonstrated to minimize correlation between independent variables for problems with less than thirty design variables. Jimenez and Balestrini generated polynomial response surface equations for a supersonic transport using a latin hypercube with 78 dimensions [223]. Polynomial RSEs are generally valid for the approximation of many engineering processes; however, they do not work well for highly nonlinear or discontinuous responses.

C.8.0.2 Neural Network Surrogate Models

An artificial neural network is "an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation" [16]. The fundamental idea in this connectionist approach is that the computational elements themselves are very simple, but like biological neurons in human brains, the connections between the neurons define very complex behaviors and computational abilities.

The technique can trace its origin to a 1943 article by neurophysiologist Warren McCulloch and mathematician Walter Pitts entitled "A Logical Calculus of Ideas Immanent in Nervous Activity" [297]. As in biological systems, a single neuron can be connected to many other neurons to create very complex networks of structures. Artificial neural networks have found widespread application in pattern recognition and classification, control processes, speech recognition, optical character recognition, autonomous robots, and the development of adaptive software agents. Their ability to model processes also makes them ideal for regression tasks, especially those with discontinuous or highly non-linear responses. Johnson provides an overview of how to use neural networks with the JMP® statistical package [224]. Other introductory works include references [337] and [383].

Although there are many types of neural networks including stochastic neural networks, radial basis functions, and committees of machines, the most common type of neural network and the technique used with success in the modeling of systems-of-systems is a feedforward neural network [141]. This type consists of several layers of interconnected neurons. Typically, three layers are used: the input layer, the hidden layer, and the output layer, shown in Figure 184. As noted in the figure, a single response has a given number of inputs, X_n, and an unknown number of hidden nodes, H_m, whose optimum configuration is problem dependent. This number can be found using numerical optimization.

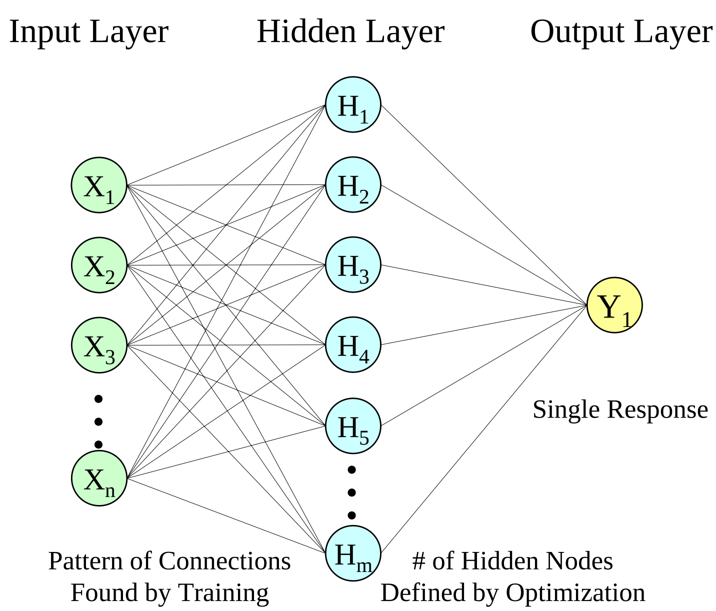


Figure 184: Typical Structure of Layers in a Feedforward Neural Network.

The power of neural networks comes from its ability to model non-linear behaviors. This is often accomplished through the use of a sigmoid curve as the transfer function between the input layer and the hidden layer. Also called the "squish" or "squash" function, the sigmoid reduces the neuron's activation level to the range of [0,1]. As an added benefit, the sigmoid function has a very simple derivative, which is needed to perform the back-propagation feature during the training process[8]. A step function or hyperbolic tangent function can also be used in place of the logistic sigmoid equation, shown in Equation 26.

$$S(z) = \frac{1}{1 + e^{-z}} \tag{26}$$

This transfer function is used to calculate the numerical value of each of the hidden nodes by application through a linear function of the input variables as shown in Equation 27.

$$H_j = S\left(a_j + \sum_{i=1}^{N} (b_{ij} X_i)\right) \tag{27}$$

Where: a_j is the intercept term for the j^{th} hidden node

b_{ij} is the coefficient for the i^{th} design variable

X_i is the value of the i^{th} design variable

H_j is the value of the j^{th} hidden node

and N is the number of input variables

The response is found using a linear function applied to the value of the hidden node as shown in Equation 28. The equation shown is for the general case of k responses. Coefficients c and d are scale factors that represent the intercept and a scalar on the interval [0,1] respectively.

$$R_k = c_k + d_k \left[e_k + \sum_{j=1}^{N_H} (f_{jk} H_j) \right] \tag{28}$$

[8] Without this non-linearity, neural networks are reduced to linear matrix multiplication problems.

Where: c_k is the response scaling intercept term for the k^{th} response

d_k is the response scaling coefficient for the k^{th} response

e_k is the intercept term for the k^{th} response

f_{jk} is the coefficient for the j^{th} hidden node and k^{th} response

H_j is the value of the j^{th} hidden node (Equation 27)

and N_H is the number of hidden nodes

Equations 26 through 28 can therefore be combined to develop a general form of the neural net equation:

$$R_k = c_k + d_k \left[e_k + \sum_{j=1}^{N_H} \left(f_{jk} \left(\frac{1}{1 + e^{-\left(a_j + \sum_{i=1}^{N} (b_{ij} X_i) \right)}} \right) \right) \right] \tag{29}$$

Where: N is the number of input (design) variables

X_i is the value of the i^{th} design variable

H_j is the value of the j^{th} hidden node (Equation 27)

a_j is the intercept term for the j^{th} hidden node

b_{ij} is the coefficient for the i^{th} design variable

c_k is the response scaling intercept term for the k^{th} response

d_k is the response scaling coefficient for the k^{th} response

e_k is the intercept term for the k^{th} response

f_{jk} is the coefficient for the j^{th} hidden node and k^{th} response

and N_H is the number of hidden nodes

That is, for k responses R and N design variables X, the neural network equation can be developed by selecting a number of hidden nodes N_H and determining values for the unknown scaling coefficients a, b, c, d, e, and f over the limits of summation illustrated in Equation 29. The process by which the scaling coefficients are determined is called *training* the neural network, and typically occurs through *back-propagation* of errors through the structure of neurons called a *network*. The training process can be time intensive due to the large number of unknown coefficients. Johnson developed a technique using the MATLAB® neural networks toolbox [128] that utilizes optimization algorithms to maximize the R^2 value of the neural network equation by manipulating the number of hidden nodes and the values of the scaling coefficients [225]. Using Johnson and Schutte's Basic Regression Analysis for Integrated Neural Networks (BRAINN) tool, a user can also specify the training time allowed or the number of discrete training attempts at each node. In this manner, the optimum configuration of the neural network equation (Equation 29) can be determined with minimal operator intervention. This training and optimization process is summarized in Figure 185. Inside the tan box, the training process guesses initial values of the coefficients of the neural network equation (Equation 29. For a given number of hidden nodes, the evaluation of this equation with a given set of input values, X, lead to an estimated value of the response, R. The algorithm compares the estimated R value to the actual value from the output of the DOE runs and evaluates the error of the prediction. On the next iteration, the training algorithm alters the coefficients and attempts to minimize the error in the evaluated response. After this error has been reduced below a user-defined threshold or the optimizer can make no more progress, the training process is complete. Within an optimization process like BRAINN, the assumed number of hidden nodes is varied and the training process repeats. BRAINN stores the results of all training passes for each user defined number of hidden nodes and after a user-specified time frame, saves the "best" equation formulated to date.

Multimodal and discontinuous behaviors can be captured well using neural network response surface equations. While any surface can be approximated using this technique, the equation for neural networks can be difficult to interpret as the standard form differs greatly from the relatively simple polynomial response surface equation. Furthermore, the selection of the number of hidden nodes is often problem dependent and can be difficult without an optimizer: too few nodes incorrectly captures the behavior of the code while too many leads to overfit problems [116]. Ender advocates generating random cases to be used as a validation set [141]. This process is analogous to the determination of model

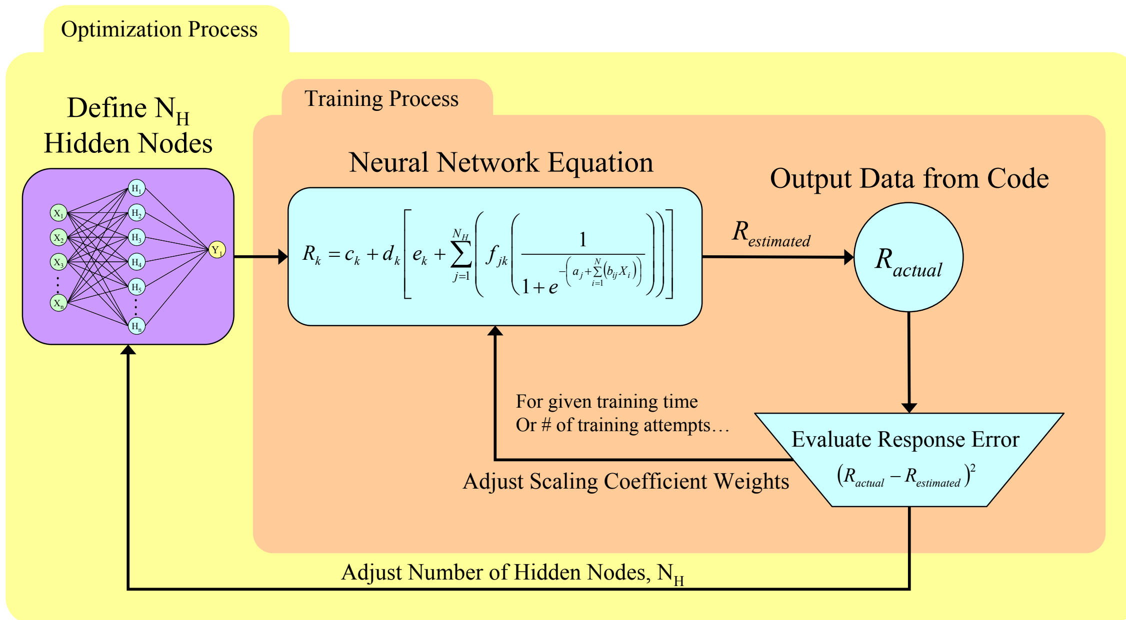


Figure 185: Training and Optimization Process for Neural Networks Using BRAINN.

representation error for an RSE. Random cases should *not* be used to develop the neural network equation as this leads to increased error from high independent variable correlation. Johnson recommends the use of a Latin Hypercube design to minimize correlation combined with a two-level fractional factorial design to capture the extremes of the design space and eliminate extrapolation of the model [224]. Initial research suggests that the goodness of fit procedure detailed by Kirby and Barros [55, 240] is also valid for neural network response surface equations.

Many other resources exist on the theory and application of neural networks to a wide array of problems. The 1988 DARPA Neural Networks Study [17] details many of the uses and applications of neural networks. Ender summarizes the theory behind artificial neural networks [141] and Johnson and Schutte provide and overview of how neural networks can be generated using the JMP® software [224] and the BRAINN module [225].

C.8.0.3 Comparison of Polynomial and Neural Network Response Surface Equations for Military Campaign Analysis

Neural network response surface equations are an enabling technology for the replacement of military modeling and simulation tools with surrogate models. This is the result of the extreme non-linear and discontinuous behaviors exhibited by such models. Due to the immaturity of the neural network technique at the time, Soban used polynomial response surface equations to approximate the campaign code ITEM [377, 378]. This was statistically valid due to the short time frames and modest number of assets in the simulation and is analogous to the linearization over small ranges. While polynomial response surface equations are a tried and true method that is applicable to a wide array of problems, military simulations over large time scales with a wide variety of assets and discontinuous changes in tactics and technologies requires the use of neural networks as surrogate models.

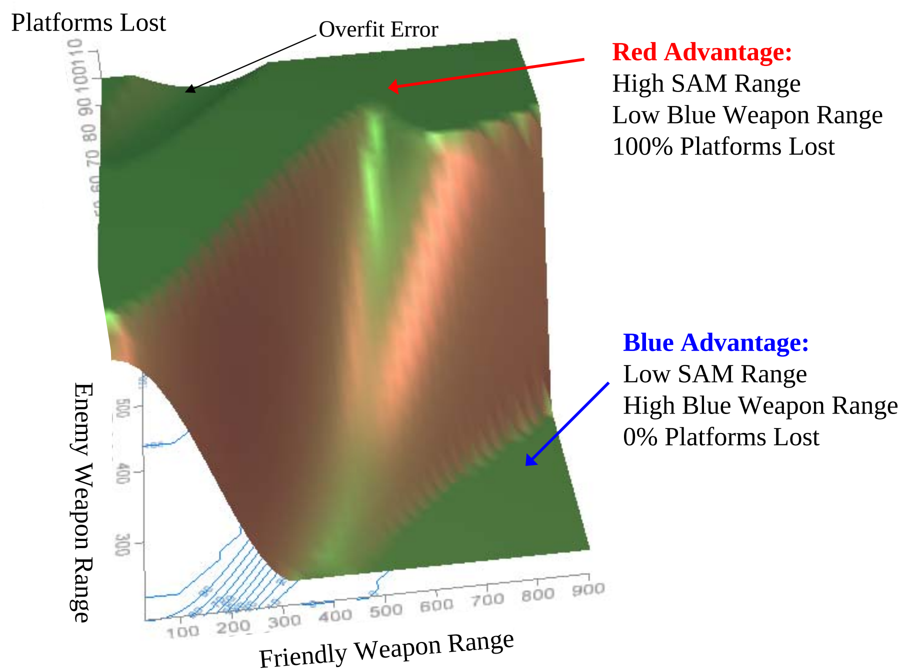


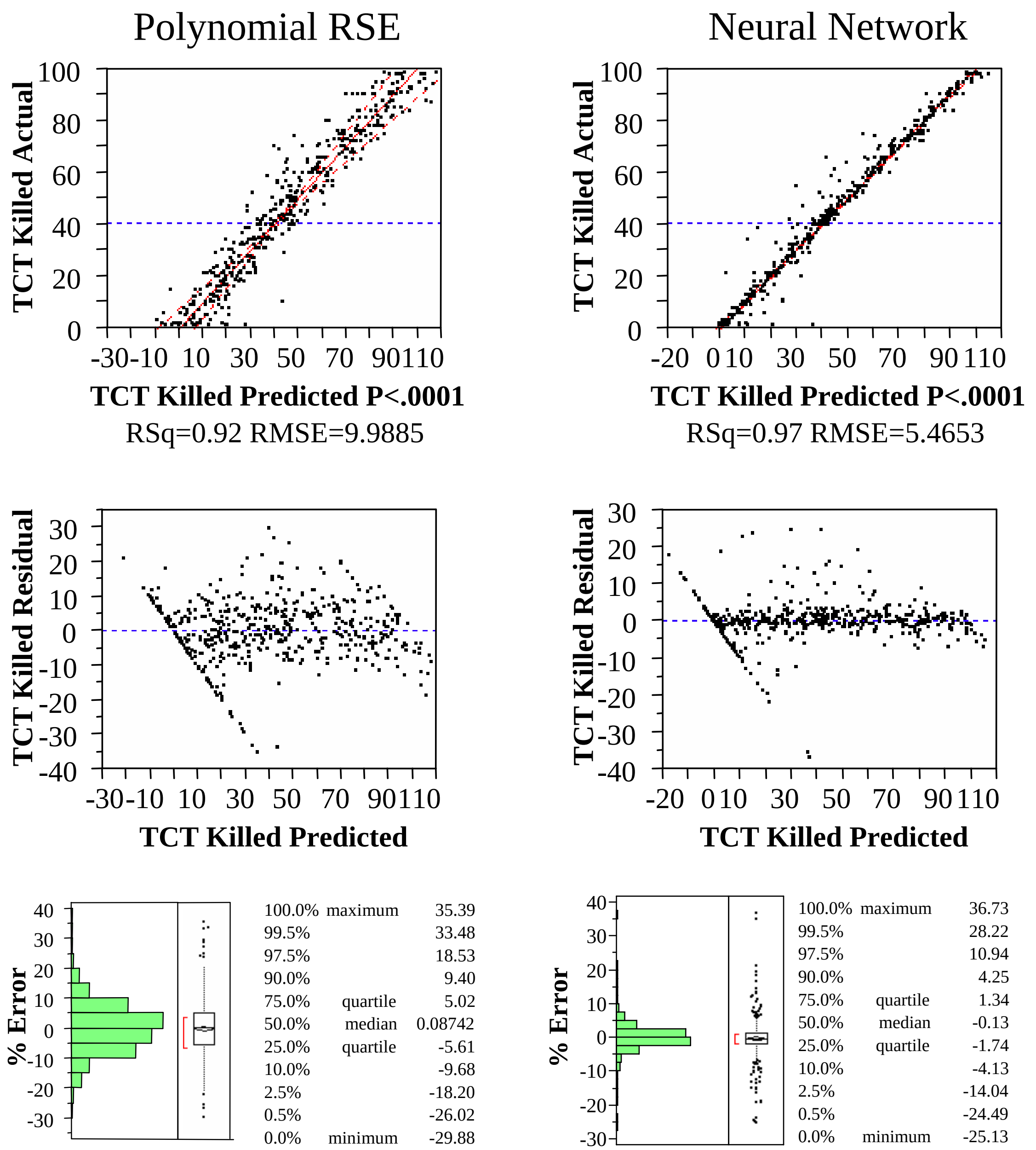
Figure 186: Example of Three Dimensional Neural Network Model Showing Multimodal Behavior.

Preliminary results from a campaign code written by Ender and Cole [289] using the methodologies of Reference [51] were analyzed using both techniques. This MATLAB® campaign tool features a notional country populated by SAM batteries and TCTs. Platforms, departing from an airbase some distance from hostile territory fire powered weapons at the targets. When all available degrees of freedom are active, the design space is highly non-linear. For example, all asset ranges and speeds are variable. When the blue force weapon range is greater than that of the hostile SAMs, the blue forces are invincible and vice versa. When TCT dwell time, used to simulate TCT “shoot-and-scoot” tactics, is set low enough blue forces cannot fly out and attack targets before they disappear, making the red TCTs invincible. This results in the multimodal behavior shown in Figure 186. The overfit penalty for the neural net is evident in the upper left corner of the figure as the plateau “dips” by several percent. This is likely due to the small number of cases run at this extreme and can be remedied by better population of this area of the design space.

To further illustrate the benefits of neural networks for system-of-systems design of military systems, a direct comparison of the two techniques is shown in Figure 187. A second order polynomial with no transformations or higher-order terms is compared to a neural network with 15 hidden nodes. Comparing the actual by predicted plots at the top of the figure shows that in general, the neural network fits the data better than the polynomial RSE. This is also manifested in a higher R^2 value, a less scattered residual by predicted plot, and a narrower distribution of error. Both models are plagued by several outliers, meaning that more data samples may be needed in regions with high residual error to enable either technique to better understand the character of that region. Also, in the case of neural network equations, it was found that sometimes large outliers are present when one of the terms in the denominator of one of the hidden node equations is near zero. Slightly changing the input parameters remedies this problem.

It is important to note that a large number of *scenario parameters* are actually included in the neural network equation. These parameters have the potential to completely change the outcome of the campaign analysis and contribute greatly to the error in both models.

While the variable transformation procedure advocated by McDonald [298] and the addition of higher order terms may improve the fit of the polynomial equation, it never correctly approximates the discontinuous behavior captured by the neural network equations. Before the advent of more complex models, a typical way to address this issue involved stitching together multiple polynomial equations at the transition points between discrete behaviors. Unfortunately, this technique only works when the location of the “knee in the curve” is known. In the study of complex systems, the location of these inflection points may be a characteristic of the variable settings at a given instant. For these reasons, neural networks



RSquare	0.921019
RSquare Adj	0.896286
Model Fit Error Mean	**3.834e-13**
Model Fit Error Std. Dev.	8.7165278
Observations (or Sum Wgts)	500

RSquare	**0.969012**
RSquare Adj	**0.96895**
Model Fit Error Mean	-0.206905
Model Fit Error Std. Dev.	**5.4719083**
Observations (or Sum Wgts)	500

Figure 187: Comparison Between Polynomial and Neural Network Response Surface Equations for a Military Campaign Analysis Code.

are an appropriate surrogate modeling technique for large simulations of systems-of-systems over reasonable time frames.

Hypothesis 4.9: *Neural Network surrogate models are appropriate for this class of problems because they can capture discontinuities endemic to systems-of-systems.*

C.9 Hypothesis 4.10: A Space-Filling Design is Ideal for Design Space Sampling

A Design of Experiments (DOE) is "a systematic, rigorous approach to engineering problem-solving that applies principles and techniques at the data collection stage so as to ensure the generation of valid, defensible, and supportable engineering conclusions" [318]. This statistical technique is concerned with selecting experiments to be performed that generate the maximum amount of data with the minimal expenditure of time and money. It originated in 1918 when the director of the Rothamsted Agricultural Experiment Station in the United Kingdom hired statistician Ronald A. Fisher to analyze historical records of crop yields. The station "had records extending over decades for crop yields from extensive plots of land each of which was treated with the same particular fertilizer" [71]. Additionally, they had records of temperature, rainfall, and other environmental factors over the same time period. This data, collected in a haphazard manner, did not answer some critical questions despite the analysis technique applied. Fisher invented the design of experiments to standardize the process by which data is collected for analysis [152, 153]. Experimental design techniques have also been refined by Yule [489], Box, and Hunter [71], Scheffé [367], Cox [102], and Taguchi [391].

C.9.1 Applications of DOEs

DOE is used for four general applications in engineering: comparison, optimization, screening/characterization, and modeling. Comparative DOEs can also be referred to as "one-factor-at-a-time" experiments. These efforts are often time consuming and a single trial typically yields only one piece of information. Comparative DOEs are the least efficient technique for exploring a design space. Optimization DOEs are concerned with determining the optimal settings of the process factors to optimize the behavior of the response.

The work of Taguchi was heavily focused on optimization [391]. While optimization DOEs are a means to determine the optimum value, they are not as useful as modeling DOEs which can also be used to calculate sensitivities for the responses and provide transparency throughout the design space.

The next two techniques have much greater influence on the design of systems-of-systems. Screening or characterization experiments are primarily used to determine what factors are important in a given process, that is, differentiate the significant few from the insignificant many. A “screening test” uses the Analysis of Variance (ANOVA) technique in combination with a two-level DOE to identify statistically significant impacts on the variability of a given response (see Section C.10.1). Screening DOEs are useful in much of engineering research and are especially critical in the design of systems-of-systems: it is important to know which factors contribute little to the variability of the response so they can be defaulted in the analysis. Modeling DOEs are used to replace the experimental process with a mathematical function that has thorough predictive power (good fit throughout the design space) and high accuracy across the range of parameters specified. Modeling DOEs are the most effective of the four techniques for system-of-systems design as they allow a complex behavior, in most cases simulated by a suite of computer codes, to be approximated by an equation. In summary, modeling DOEs are used to create response surface equations (RSEs).

C.9.2 Types of Sampling Techniques (DOEs)

There are several types of Designs-of-Experiments used in the screening and modeling roles. According to Montgomery, there are eleven features that impact the quality of a design including a “reasonable distribution of data points throughout the region of interest,” minimization of the number of experiments performed, and good predictability in the estimation of model coefficients [55, 300]. Since different sampling techniques may have better applicability to certain classes of problems, several different types of DOEs, illustrated in Figure 188, are examined herein.

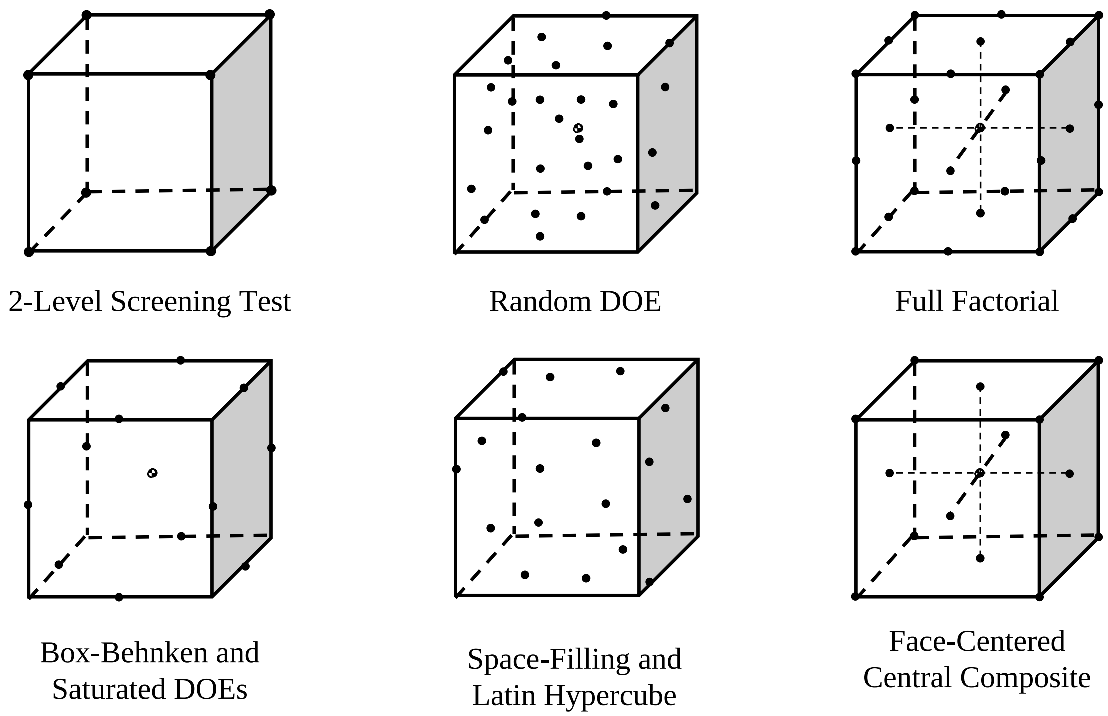


Figure 188: Types of Experimental Designs for Screening and Modeling Applications (Adapted from [55]).

Screening DOEs are almost exclusively composed of two-level orthogonal arrays. These can be seen as combinations of on-off switches that test different combinations of parameters at their high and low values. The simplest kind of DOE to generate is a full-factorial two-level DOE which requires 2^n experimental runs and leaves no degrees of freedom for error. The number of experiments can easily be reduced using a fractional factorial design: "a factorial experiment in which only an adequately chosen fraction of the treatment combinations required for the complete factorial experiment is selected to be run" [43]. Properly chosen fractional factorial designs should be both balanced (all treatment combinations have the same number of observations) and orthogonal (the sum of the products of each vector in the design is zero) [318]. Balanced, orthogonal arrays have the property of low independent variable correlation. They are typically of Resolution III, meaning the main effects are confounded with two-variable interactions. For this reason, two-level fractional factorial experiments are not appropriate for generating polynomial statistical models of order 2 or higher as they can only model linear behavior.

The “simplest” type of modeling DOE, a randomly generated DOE, is not really a DOE at all since it is not created by a structured process. Random DOEs consist of points stochastically distributed about the design space or a subset thereof. While these designs are easy to create, they suffer from a high degree of independent variable correlation. Random designs are usually not used to create models, but are good for examining the model representation error of surrogate models [55].

Full factorial designs are a classical experimental design and a brute force approach that essentially runs all combinations of variables at the levels specified. A three-level full factorial design therefore requires 3^n cases while a five-level design requires 5^n. The curse of dimensionality is especially evident in full factorial designs. Although the run time is large, the extremes of the design space are well covered by this technique although the interior of the design space is only sampled through the introduction of more levels and hence drastically increasing the number of cases required. Full factorial designs are considered to have infinite resolution as they have no confounding if the appropriate number of levels is chosen for the degree of the model required [318].

Box-Behnken designs, proposed by G.E.P. Box in 1978, are independent quadratic designs that do not contain embedded factorial or fractional factorial designs [318]. These designs are similar in construction to the two-level screening test designs by Plackett and Burman. Box-Behnken designs are formulated by combining two-level factorial designs with incomplete block designs. They suffer from a complex aliasing structure (most are Resolution IV); however, they are efficient designs from the standpoint of run time. These designs sample the edges of the design space as opposed to the corners, and therefore response surface models using Box-Behnken designs suffer from extrapolation problems at the extremes of the design space which leads to significant error. For this reason, these designs are typically avoided except in situations where the modeling tool requires extreme run times and it is unlikely that corner point solutions will be considered.

D-optimal designs or saturated DOEs are non-orthogonal designs with correlated effect estimates that represent the minimum number of cases that can be executed to develop a model at a desired order [318]. Generation of D-optimal designs for optimality involves maximization of the determinant of the Fischer information matrix $[X^T X]$. D-optimal designs generally result in poor coverage of the design space and should only be used when the standard types of designs require too many runs or the design tools are known to fail at the ranges required for the aforementioned designs.

A central composite design (CCD) is a fractional factorial design of Resolution V which combines a two-level full-factorial design with "star points" located at a defined distance from the center point. The location of the star points defines whether the design is inscribed, face centered, or circumscribed. Face-centered CCD's have been used widely for aerospace applications [55, 240]. Interior sampling is absent save a single center point, but the extremes of the design space are well-covered. As previously mentioned, this has advantages from a modeling approach but can result in unconverged designs if the simulation tool generating the responses cannot run when two variables are at their extreme values. Creation of CCD's for large numbers of independent variables (over 20) is a computationally intensive task as the computational algorithm seeks to create rows of inputs that have no correlation to any other row. To generate designs in a reasonable time frame it is often necessary to place a threshold on the allowed correlation; however, the correlation generally becomes excessive in central composite designs with more than approximately 30 independent variables.

Space-filling designs, which literally fill a n-dimensional space, "should be used when there is little or no information about the underlying effects of factors on responses" and are "useful for modeling systems that are deterministic or near-deterministic" such as computer simulations [275, 366]. While random points can be used to fill a space, an alternative scheme called "sphere-packing" is used to minimize the maximum distance between any two points in n-dimensional space, akin to placing billiard balls into an n-dimensional box [366]. Mathematical techniques to assess this distance have been developed extensively in the literature [147, 227, 485]. "A good space-filling design is one in which the design points are scattered throughout the experimental region with minimal unsampled regions; that is, the voided regions are relatively small" [96]. As a result, space-filling designs can be effective for neural network models when the exact location of inflection points in the design space is unknown.

Finally, Latin Hypercube (LHC) designs, developed by Ronald L. Iman, J. C. Helton, and J. E. Campbell in the early 1980s, attempts to distribute points evenly through the design space using a combination of uniform designs[9] with a sphere-packing scheme [215]. The primary advantage of the LHC technique is that the number of runs required in the design can be defined by the user depending on the density of coverage of the design space desired. The LHC algorithm creates the desired number of runs, maximize the coverage of the design space, and minimize independent variable correlation. LHC designs can be made orthogonal (the entire sample space is sampled evenly); however, efficient orthogonal sampling is difficult in practice because all random samples are generated simultaneously. Generally, it is easier to specify a threshold for the allowed independent variable correlation to minimize the error in the final design. Like other space-filling techniques, LHC's have the benefit of a rich sampling of the interior though they may be subject to high independent variable correlation if improperly created. If a small number of runs is used, they do not adequately cover the extremes of the design space; however, they do not suffer the same correlation problems as CCD's for a large number of independent variables. A latin hypercube design was used by Jimenez and Balestrini to generate response surface equations for a supersonic transport using forty design variables with reasonably low correlation [223]. One of the major disadvantage of a LHC is that the creation of designs with the uniform constraint is much more time consuming than the sphere-packing scheme, as Fang and Wang note, the number of readily available uniform designs is severely limited by the difficulty in creating designs for many combinations of variables and runs [96, 147].

When modeling very complex nonlinear behaviors, Johnson advocates the combination of latin hypercube designs with a fractional factorial design that extends to the extremes of the design space [225]. Biltgen, Ender and Cole demonstrated that this technique is viable for a system-of-systems formulation [289]. Since the uniform design constraint, which limits the number of trials at a given level to one, is not an important aspect of experimental designs needed for this work, a space-filling design is used. This assertion is supported

[9] A uniform design "minimizes the discrepancy between the design points (which have an empirical uniform distribution) and a theoretical uniform distribution" [366].

by Dr. Christopher M. Gotwalt, Senior Research Statistician with the SAS Institute, who notes that a pure space-filling design is most appropriate for this type of computer simulation [182]. When necessary, a central-composite design can be used to supplement the space-filling design; however, for the large number of runs considered in this work, coverage of the design space generally extends to the extremes without this supplemental DOE.

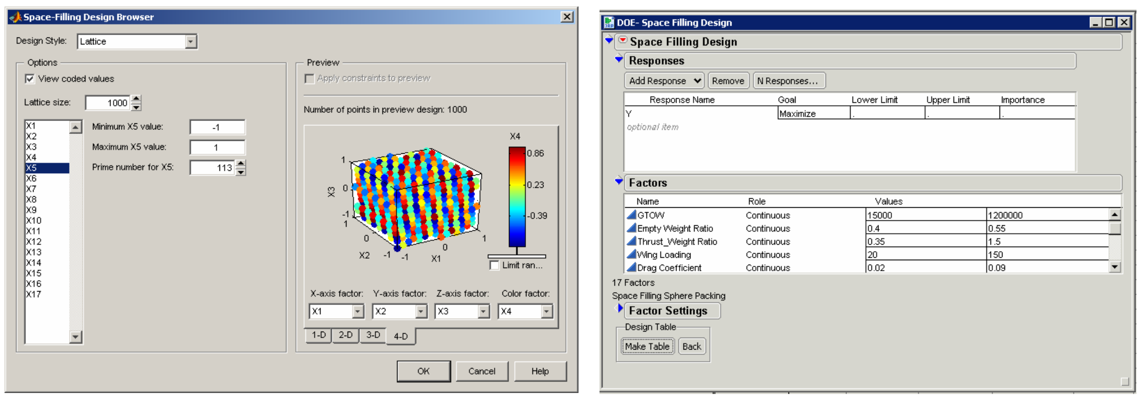


Figure 189: The MATLAB® Model Based Calibration MBC Toolbox (left) and the JMP DOE Creation Tool (right) [274].

Graphical representations of the sampling techniques described above are shown in Figure 188 and the benefits and drawbacks of the techniques are generalized in Table 33. Many computerized tools exist for the creation of experimental designs including Design Expert, JMP® [19], and the MATLAB® Model-Based Calibration Toolbox (MBC)[274]. Both the MATLAB® MBC toolbox and JMP® (shown in Figure 189) are used in this research based on ease of use, simplicity, and speed.

Hypothesis 4.10: *For non-linear systems-of-systems problems a space-filling design provides excellent coverage of the design space and facilitates the creation of neural networks.*

Table 33: Characteristics of Experimental Designs for Modeling Applications.

Type	Pros	Cons	Num. of Cases for 10 Vars.
2-Level Screening Test	Used for ANOVA Good for Narrowing the Variables Used	Not valid for modeling anything but linear processes	Less Than 1024
Random	Simple to Create Analysis can be stopped at any time	High correlation between cases	User Defined
3-Level Full Factorial	Excellent coverage of design space Error reduced through excessive sampling	Brute Force. Large Number of cases Curse of dimensionality evident	59,049
Box-Behnken	Pre-determined given num of variables Avoids corners of the space (avoid crashing analysis tool)	Heavy use of extrapolation in model due to small number of cases Avoids corners of the space (could eliminate valid designs)	
D-Optimal (Saturated)	Minimum number of runs for modeling DOE	Heavy use of extrapolation in model due to small number of cases	66
Central Composite	Extremes of design space considered Little use of extrapolation	Use of corners can crash analysis tool Generation of large CCD's requires extensive computing power	1045
Space-Filling	Good sampling of interior of the design space Requires little computational time to create	Can be correlated if incorrectly made May not adequately cover the extremes of the design space	User Defined
Latin Hypercube	Benefits of space-filling + uniform	Uniform constraint lengthens creation time	User Defined

C.10 *Hypothesis 4.11: Sensitivities Can be Evaluated Using the Prediction Profiler and the ANOVA Technique*

"Metallurgist: a pseudo scientist, who uses undetermined suppositions, indefinite theories, and inexpressible hypotheses; which are based on unreliable information uncertain quantities, and incomplete data; derived from non-reproducible experiments and incomplete investigations; using equipment and instruments of questionable accuracy, insufficient resolution, and inadequate sensitivity, to arrive at timid, tentative cloudy, abstruse, and non-committed conclusions prefaced by the phrase, 'IT DEPENDS' "

Attributed to Douglas J. Robinson

The above quote, present on the cubicle wall of every materials engineer in the United States, is just as suitable for systems-of-systems engineers as it is for metallurgists. The phrase "IT DEPENDS" underscores the sensitivity of conclusions on assumptions. Since in practice it may be difficult to truly *answer* an analysis question due to the confounding complexity of the problem, this section examines techniques for assessing the sensitivity of key metrics to input variables under the designer's control.

C.10.1 Analysis of Variance

Analysis of Variance, or ANOVA, is a statistical procedure that aims to identify source of variability in a process. This technique was pioneered by Sir Ronald Fisher [16]. Using a two-level design of experiments that runs points at the extremes of the design space, the ANOVA procedure can be performed and visualized using a *Pareto Chart*. Named after Vilfredo Pareto, an Italian, who made the observation in 1906 that 20% of the population owned 80% of the property, the "Pareto Principle" was later generalized by Joseph M. Juran to state that 80% of the variability of a process generally comes from 20% of the causes [16]. This "80-20 rule" is rather a guideline that is problem dependent. A pareto chart is a type of histogram that is useful for visualizing the impact of variability on a given response. A pareto chart for a military campaign simulation is shown in Figure 190.

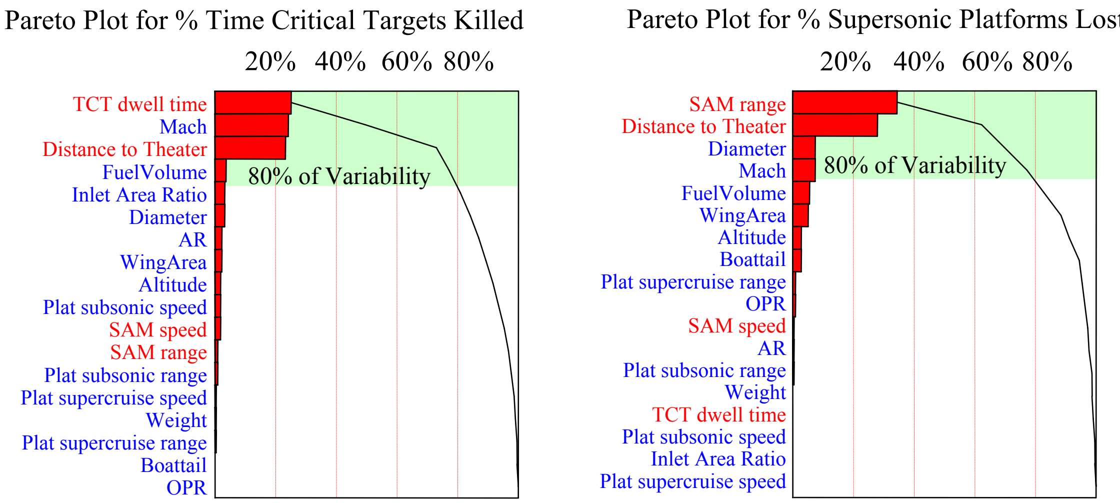


Figure 190: Illustrative Pareto Chart for Military Campaign Simulation.

The magnitude of the bars in Figure 190 illustrates the relative contribution of that parameter to the overall variability of the selected response. The curved black line sums the cumulative impact of the variables above that point and the green shaded region highlights the variables which contribute to 80% of the variability of the response. The color codings on the variable names indicate whether the parameter is under control of the friendly (blue) player or the adversary (red) player.

It is important to note that the ANOVA procedure is valid for the variables selected *and the ranges those parameters are allowed to vary.* In this example, the importance rankings illustrated in the Pareto chart of Figure 190 are also a function of scenario parameters such as force ratio (ratio of blue to red forces), percentage of time-critical targets on the battlefield, and ratio of subsonic to supersonic platforms on the blue side. The green shaded region represents the 80% threshold. Coincidentally, four variables compose 80% of the variability for both responses. Variable names in blue represent design variables for the missile, while red shaded names indicate noise parameters that are a function of the threat situation in the scenario and cannot be defined *a priori*. Pareto charts depict variability of the response. When the positive or negative variability on the response is depicted, this is popularly called a "tornado chart." For the parameter settings and ranges used, the most critical parameters for killing time critical targets are the amount of time the target is on the battlefield, the cruise Mach number of the weapon, and the distance from the airbase to the region of interest. The plot of blue supersonic platform losses indicates that SAM range and distance to theater are the two primary factors, followed by characteristics of the weapon. Over the range of variability provided, weapon design variables far outstrip those of the platform. If any scenario parameters are changed, the outcome of this plot is completely different, as scenario assumptions have a very large impact on the variability of the response.

The ANOVA procedure the the Pareto chart are useful for determining potential causes for observed behavior; however, since the range of the input variables plays a role in the factor's evaluation in the Pareto chart, it is difficult to generalize cause and effect relationships. All observed trends are valid *for a specific scenario and its underlying assumptions*. The static nature of the Pareto chart drives the need for a more dynamic sensitivity analysis environment.

C.10.2 Sensitivity Analysis Using the Prediction Profiler

One of the primary advantages to surrogate models is the ability to perform parametric sensitivity analysis and view trends across the range of design variables. A graphical technique inherent to JMP® called the *prediction profiler* is a useful way to visualize these sensitivities by changing one variable at a time and observing a change on the responses. The prediction profiler evaluates the response surface equations and displays curved lines called the *prediction trace*: "the predicted response as one variable is changed while the others are held constant at the current values" [18]. An example of the prediction profiler is shown in Figure 191. The vertical red lines indicate the current value of the X variables while the horizontal lines correspond to the value of the responses. The individual cells can also be interpreted as a the partial derivative of the response (Y-location) with respect to a given design variable (X-location) with all other design variables held constant. Moving the hairlines evaluates the response surface equations at new values, acting as a "calculator" for real-time design space exploration and optimization. Also, the slope of each prediction trace is analogous to the sensitivity of the response on a change in the design variable.

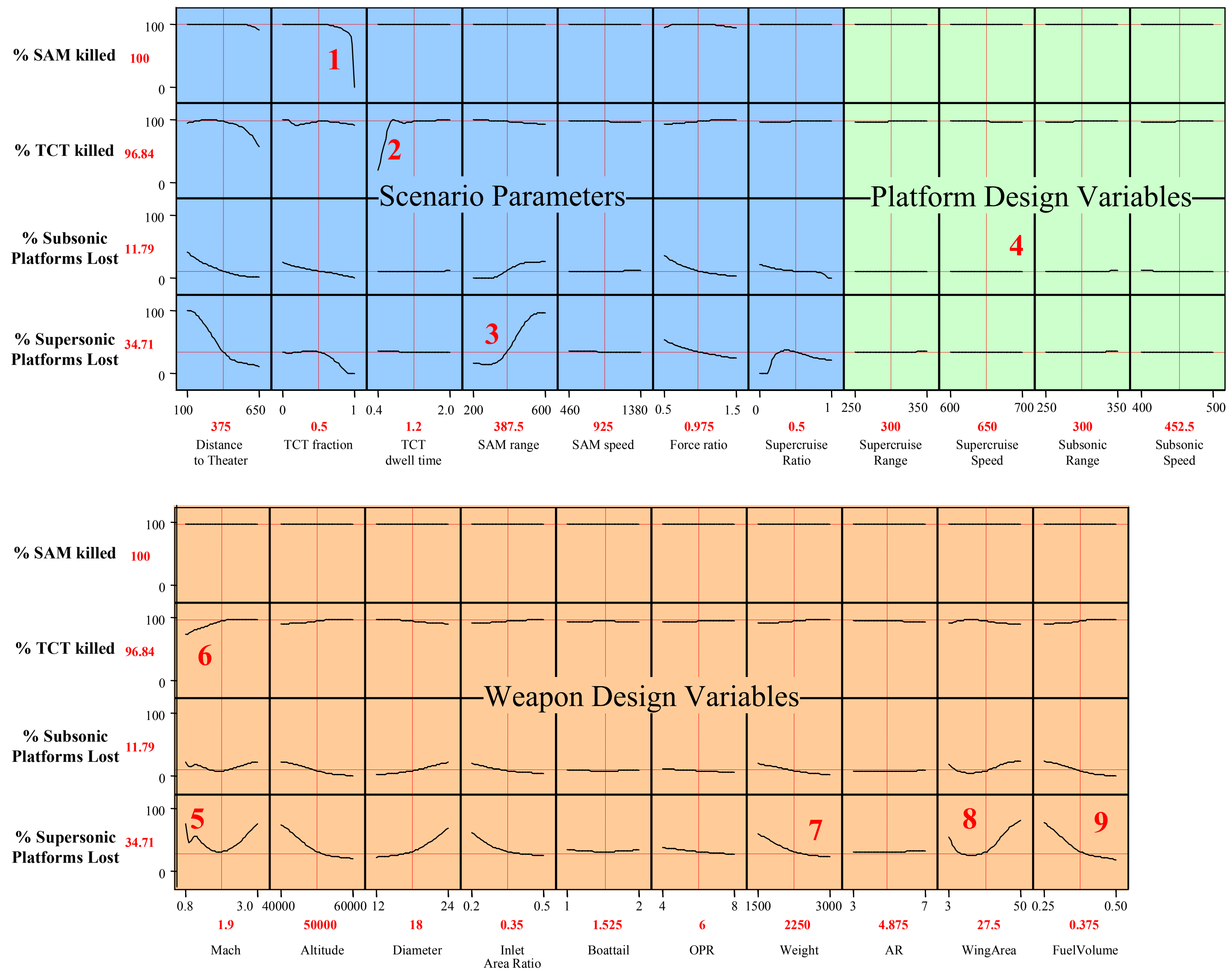


Figure 191: Prediction Profiler for a Military Campaign Analysis Code.

This can be used to determine the required code fidelity for a given input: steeper traces indicate that the penalty for errors in a computational tool has greater overall impact on the responses.

The prediction profiler shown in Figure 191 was generated using neural network equations and therefore has some unusual non-linear behaviors. Several of these behaviors have been labeled in the figure. In the definition of scenario parameters, the percentage of SAM sites killed is greatly dependent on the TCT fraction (1). TCT fraction is defined as the ratio of time-critical targets to SAM sites. As this ratio approaches 1.0, the battlefield is comprised entirely of TCTs, therefore, there are no SAM sites to kill, represented by the precipitous drop. A similar drop-off is evident in (2). This reveals that for the other independent variable settings, it becomes increasingly difficult to kill time critical targets with a dwell time under 40 minutes. The percentage of platforms lost (3) increases dramatically as the SAM range increases. The flattening of the curve on the left indicates that the platform range is greater than the SAM range: platforms can fire weapons from outside defended airspace and are hence nearly invincible[10]. When SAM range is greater than the friendly shoot-back range, no platforms survive. Region (4) indicates that the platform design variables have very little impact on the response. This is due to their small range of variability. After examining the prediction profiler, the designer may wish to increase the range of variability of the platform design variables, as their effects are "washed out" by those of the scenario parameters and weapon design variables. The prediction trace (5) shows a slight hump in the number of platforms lost. This corresponds to a weapon cruise Mach number of 1.0, which penalizes the weapon range unfairly and thus contributes to platform deaths because the platforms have to fly farther into defended airspace before releasing weapons. The trace (6) shows that weapons with a Cruise Mach greater than approximately 2.0 kill all time critical targets for the scenario settings specified. Trend (7) appears counterintuitive: when weapon weight increases, more platforms survive. While aerospace engineers generally try to reduce gross weight, heavier weapons carry more fuel

[10]Based on the scenario, it is possible for undetected SAM sites to shoot back. This is why the curve approaches but does not reach zero. The probability of detecting all SAM sites is never 100%.

and hence have greater range. Weapon weight (7) is correlated with fuel volume (9). Finally, the trend for weapon wing area (8) indicates that there is an optimum around 15 ft^2. This is a function of the other parameters of the missile design, but in general a missile with too small of a wing expends thrust to stay aloft while a missile with a large wing area has excessive profile drag.

In this manner, the prediction profiler can be used to debug analysis tools, discover local optima, and identify trends in the analysis that merit further examination.

Hypothesis 4.11*: The prediction profiler (essentially a matrix of partial derivatives) or the ANOVA technique can be used to ascertain what variables are significant with respect to the capability-level metrics.*

C.11 Hypothesis 4.12: Monte Carlo Simulation Can Be Used to Span the Design Space and Account for Uncertainty

Probabilistics, meaning "relating to or based on probability," dates to the Mid 17th century [347]. Originally based on correspondence between Pierre de Fermat and Blaise Pascal, the first textbook on the subject, *On Calculating in Games of Luck*, was published by Christiaan Huygens in 1657 [221]. Other contemporary contributors to probability include Jakob Bernoulli, de Moivre, Laplace, Daniel Bernoulli and others [16]. Probability theory is a wide field concerned with the liklihood of event occurrance, probability distributions, combinations and permutations, games, and many other fields. Mavris observed that uncertainty arising from ambiguous requirements, design and operational uncertainty, and unknown analytical tool fidelity can be quantified using probabilistic techniques [292].

C.11.1 Monte Carlo Methods

The most accurate technique for probabilistic analysis is the Monte Carlo method. Named after the famous Monaco casino, Monte Carlo methods were originally called "statistical sampling." The more colorful moniker was popularized by pioneers Ulam, Fermi, von Neumann, and Metropolis [16]. The technique utilizes random or pseudorandom numbers to stochastically simulate various physical and mathematical systems and has been applied to quantum physics, aerodynamics, solutions of integro-differential equations, and the development of the atomic bomb. The method became widely applied with the advent of digital computers, replacing tables of random numbers previously used for statistical sampling. While many computerized tools for engineering design are not probabilistic, a viable methodology is to encapsulate existing deterministic tools with a Monte Carlo interface [323].

A typical use for Monte Carlo methods is to create a histogram of a process by running a large number of cases using random or pseudorandom numbers. This histogram is also referred to as a probability density function (PDF). The integral of the PDF between points a and b yields the probability that a number falls between a and b:

$$P\left(a \leq X \leq b\right) = \int\limits_{a}^{b} f\left(x\right) dx \tag{30}$$

This is analogous to the area under the PDF between a and b. The cumulative distribution function, CDF, is the integral of the PDF and yields the probability that a random number is less than a threshold value x:

$$F\left(x\right) = P\left(X \leq x\right) = \int\limits_{-\infty}^{x} f\left(u\right) du \tag{31}$$

In addition to purely random number generators (uniform distribution), many different distributions can be used to characterize the inputs to a Monte Carlo analysis. The process of using a Monte Carlo simulation to create a PDF and CDF is shown in Figure 192.

In a process formalized by Bandte, a Monte Carlo simulation (MCS) over a range of input parameters can be examined in multiple dimensions and compared using a joint probability estimation approach [52, 53]. This technique is useful for comparing multiple dimensions simultaneously and looking for correlations between output parameters. Kirby applied this technique to the selection of technology portfolios across eight dimensions simultaneously [241]. A joint probability distribution is shown in Figure 193.

Monte Carlo methods can be used for integration by approximating the integral of a function by evaluating and averaging the function over a set number of points. As the number of

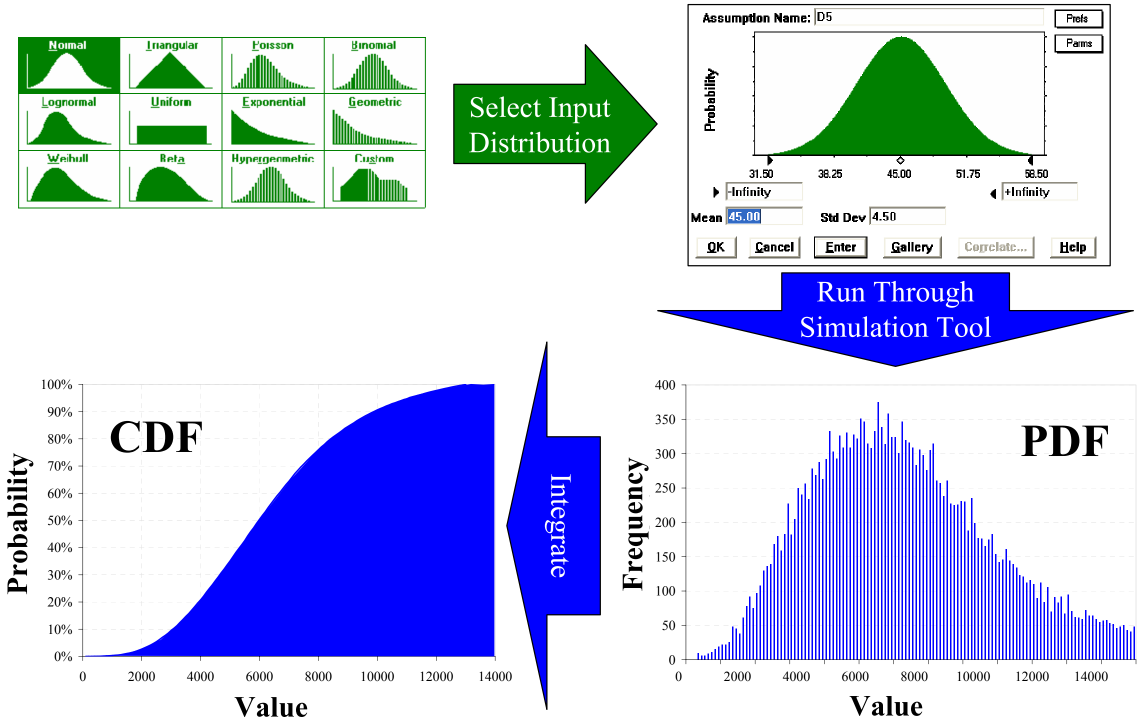


Figure 192: Using Monte-Carlo Simulation to Create Probability Density Functions and Cumulative Distribution Functions.

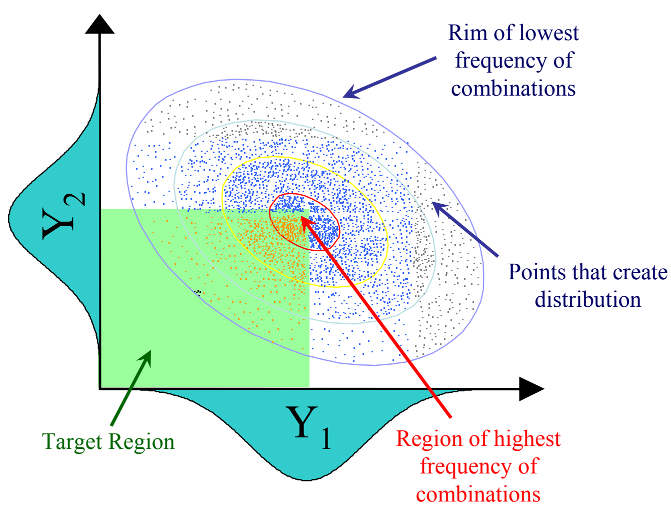


Figure 193: Joint Probability Distribution.

points approaches infinity, the integral evaluated using the Monte Carlo method approaches the true value of the integral. While numerical integration is usually straightforward in a single dimension, the Monte Carlo method is useful for multidimensional integration over problems with many degrees of freedom, where the approximation of the function f is given as:

$$\int\limits_{\bar{I^s}} f(u)\,du \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{32}$$

Where $\bar{I^s}$ is a multidimensional cube with s degrees of freedom and N is the number of elements in the sequence of random numbers.

C.11.2 Other Probabilistic Techniques

In contrast to a pure Monte Carlo method, a Quasi-Monte Carlo method replaces the set $\mathrm{X}_1 \ldots \mathrm{X}_N$ with a low discrepancy sequence, also known as a sequence of quasirandom numbers [191]. "Quasirandom numbers are generated algorithmically by computer, and are similar to pseudorandom numbers while having the additional important property of being deterministically chosen based on equidistributed sequences in order to minimize errors" [418, 480]. Morokoff and Caflisch noted that for a number of example problems, the Quasi-Monte Carlo method tended to yield more accurate results than a Monte Carlo method using the same number of points [302].

Markov Chains, which are discrete time stochastic processes have the property that the future is conditionally independent of the past, given the present. Queueing theory, statistics, population process analysis, and Brownian motion are examples of stochastic processes that can utilize Markov Chains. A random walk algorithm is another example application.

In addition to traditional approaches where Monte Carlo simulation is used with a simulation code and the more elegant approach where MCS is combined with surrogate models, another approach used in probabilistic analysis is to approximate the probability distribution function itself. "This is based on the notion that in order to obtain the cumulative distribution function, not all probability levels need to be identified" [53]. One such technique, Fast Probability Integration (FPI) is a computer program developed at the Southwest Research Institute for NASA Glenn Research Center [379]. FPI combines a mean value method with a Most Probable Point analysis to determine the CDF for a single response function. FPI is a valid approach when it is not possible to either create accurate surrogate models or run a full MCS with the physics-based code. The acceptance of this technique has been somewhat limited based on widespread limitations of the method coupled with improvements in surrogate modeling techniques and advances in computational power since 1995.

C.11.3 Uses of Probabilistics in Design

Probabilistics is a central concept in the robust design of systems and reliability analysis [257]. While both use the same techniques and methods, the focus of reliability analysis is usually in the extreme conditions of a distribution while robust design is focused on operation around a likely design condition. Probabilistic approaches have been used extensively for the analysis of systems effectiveness [378], radar cross section prediction [206], missile design [60, 63, 251], propulsion system selection [294, 358], economic uncertainty [288], and space systems [365] to name a few.

Viability of designs through analysis of uncertainty is useful in determining how a given design may be sensitive to variations in noise parameters. A typical example is the assessment of economic metrics for commercial transport when load factors, utilization, and fuel costs are used as noise variables. Probabilistics can also be used for large-scale design space exploration, useful in the design of systems-of-systems.

Hypothesis 4.12: *Monte Carlo simulations are a convenient and uncomplicated method for quantifying uncertainty that have been used successfully for this class of problems.*

APPENDIX D

SUMMARY OF ARCHITECTURE TERMS

"No person who is not a great sculptor or painter, can be an architect. If he is not a sculptor or painter, he can only be a builder."

-John Ruskin

The term *architecture* is central to the development of this work; however, the term itself has a variety of ambiguous meanings. Architecture, whose Greek roots mean "first craftsmanship," is literally "the art and science of designing structures" [22]. The balance between "aesthetic" qualities and a scientific understanding of the fundamental properties of a structure in its environment is as critical for buildings as it is for military system architectures. Architecture is governed by the principle of "form follows function," which is also a central tenet of systems engineering. There are many useful definitions of architecture:

- DoD defines an architecture as "a framework or structure that portrays relationships among all the elements of the subject force, system, or activity" [468]. This definition is ambiguous because many experts agree that an architecture is *not* a framework.
- The DoD Integrated Architecture Panel developed a more rigorous definition[1], "the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time" [136]. This statement is less confounding and establishes that architectures contain things, connections between things, and rules or standards regarding the elements in an architecture.
- Maier defines an architecture as "the structure (in terms of components, connections, and constraints) of a product, process or element" [268]. This definition is consistent with the DoD Integrated Architecture Panel definition.

[1]This definition is often erroneously credited as coming from IEEE standard 610.12. It is originally credited to Dewayne E. Perry and David Garlan; however, the location of its original citation is unknown.

- The IEEE Architecture Working Group defines an architecture as "the highest-level concept of a system in its environment" [268]. This definition is most relevant to capability-based design.
- Finally, an elegant definition proposed by Daw is "an organization of resources" [121].

The term "framework" is misleading: architectures are not frameworks. Although often listed as synonyms, in the military simulation community, the two are greatly different terms. An architecture is a collection of things. A framework is a computational (or other modeling) environment that allows analysis of an architecture. In the information systems community, frameworks are used to construct architectures. The archetypical framework is the Zachman Framework for Enterprise Architectures, which uses a grid model that asks six questions (What, How, Where, Who, When, Why) of five stakeholder groups [491, 492, 493]. The answers to these questions form the architecture. The "Zachman Framework" is shown in Figure 194.

In software engineering, a framework is "a defined support structure in which another software project can be organized and developed" [16]. Frameworks for these tasks may include Rational® Software Modeler [31] and Microsoft® Visual Studio [26]. In this research, the term framework refers to the software tool used to perform simulations.

With the above distinction between architectures and frameworks in mind, the term "Department of Defense Architecture Framework" (DoDAF) is less confusing. "The DoDAF provides the rules, guidance, and product descriptions for developing and presenting architecture descriptions to ensure a common denominator for understanding, comparing, and integrating architectures" [27]. This **framework** is used to view **architectures**. The DoDAF standard, developed by the defense acquisition community, facilitates large-scale system-of-systems design by establishing standards for the *depiction* of architectures to promote interoperability both across capabilities and between integrated architectures. Understanding the "views" of DoDAF is critical to capability-based design, network centric warfare, and planning for joint operations that require the integration of multiple architectures. The term "views" refers to different ways of looking at the same architecture and is analogous to the top, side, front, and isometric views of an engineering drawing: each view

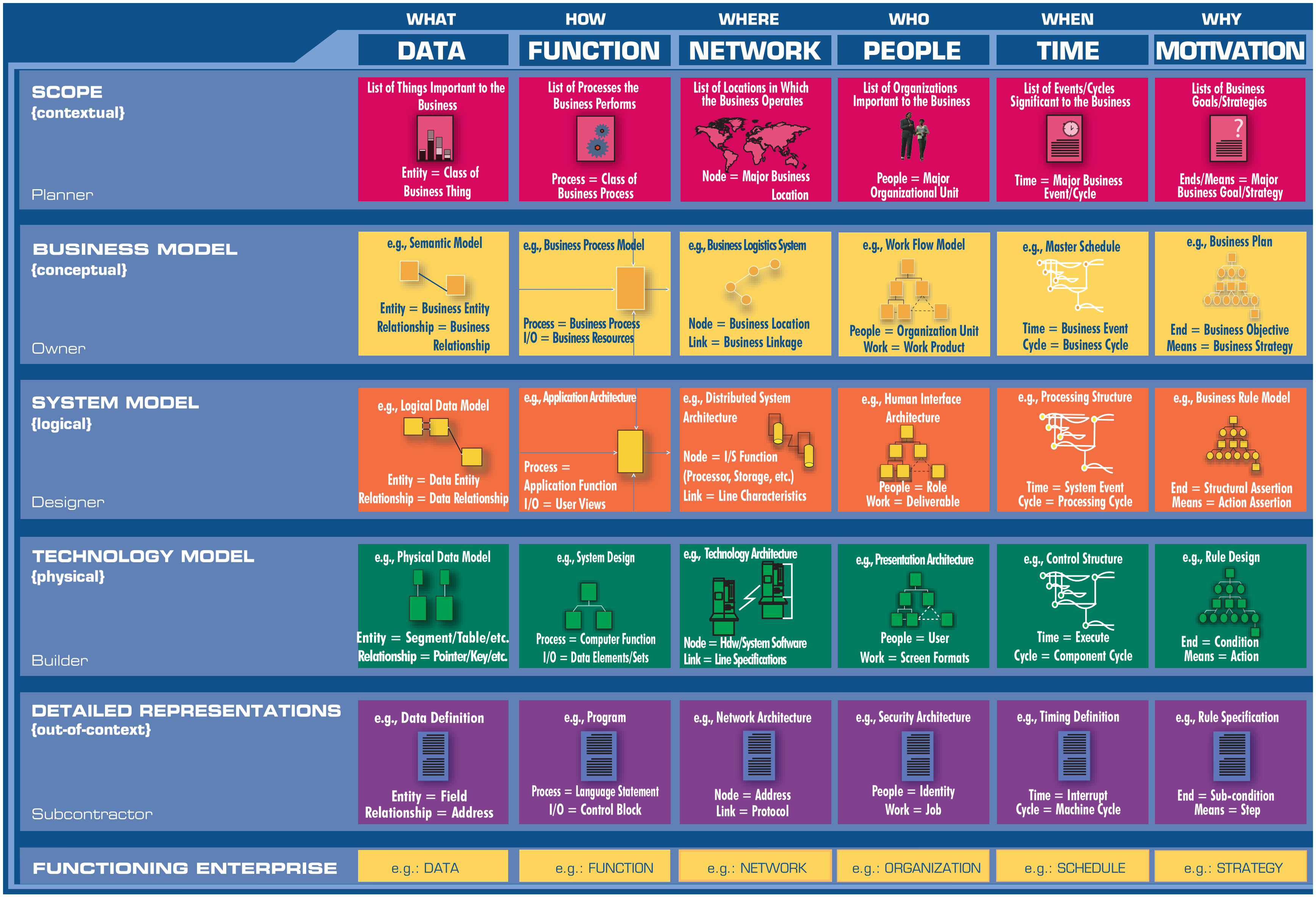


Figure 194: Zachman Framework for Enterprise Architectures [492, 390].

provides information to the user. DoDAF views are inherently hierarchical, multidimensional, and very detailed. Properly defining these views is critical to requirements definition and capability analysis.

According to Reference [27], the DoDAF views are:

- **Operational View** (OV): description of tasks and activities, operational elements, and information exchanges required to accomplish DoD missions.
- **Systems View** (SV): description, including graphics, of systems and interconnections providing for, or supporting, DoD functions.
- **Technical View** (TV): The minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specified set of requirements.
- **All View** (AV): Provides information pertinent to the entire architecture but does not represent a distinct view of the architecture. Sets the scope and context of the architecture.

The operational view can usually be determined through brainstorming and functional decomposition. It says *what the architecture does*. The systems view can be seen as a type of physical decomposition that says *what the architecture is composed of*. This could be a series of flowcharts or storyboards of what the architecture looks like. The technical view is a more comprehensive view. If the operational and systems view can be considered pre-conceptual design, the technical view is more appropriate in the preliminary design phase. This defines *how the elements of the architecture interact*. With a function, form, and method for interaction, the architecture is defined. The all view provides additional higher level information but is excluded from some conceptual design formulations. The flow down of information through the DoDAF views is shown in Figure 195.

Just as software engineers use frameworks to plan the content and structure of their code, the DoD Architecture Framework provides a planning environment for military architectures. In his seminal 2004 work, Dickerson describes a methodology for using an

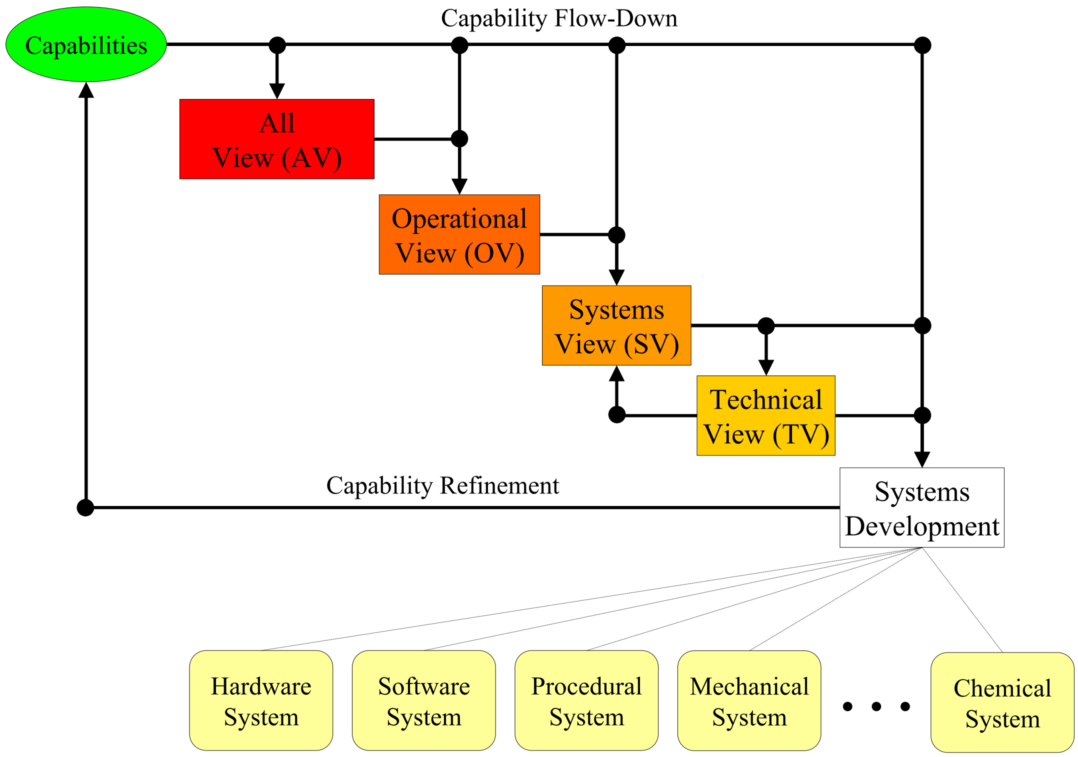


Figure 195: Flow of Information from DoDAF Views to System Development Models (Modified from Reference [349]).

architecture framework to support capabilities-based acquisition, and identifies the significance of the architecture views as follows:

> *"Just as a building architect develops blueprints so that individual contractors can determine the scope and requirements of their jobs, the systems architect develops blueprints in accordance with the DoD Architecture Framework so that individual program managers can determine the scope and requirements of their systems"* [134].

The DoDAF views depicted in Figure 195 function as the blueprints to describe the various aspects of the architecture design. Dickerson also outlines a process for using the DoDAF views for systems-of-systems systems engineering and acquisition, illustrated in Figure 196.

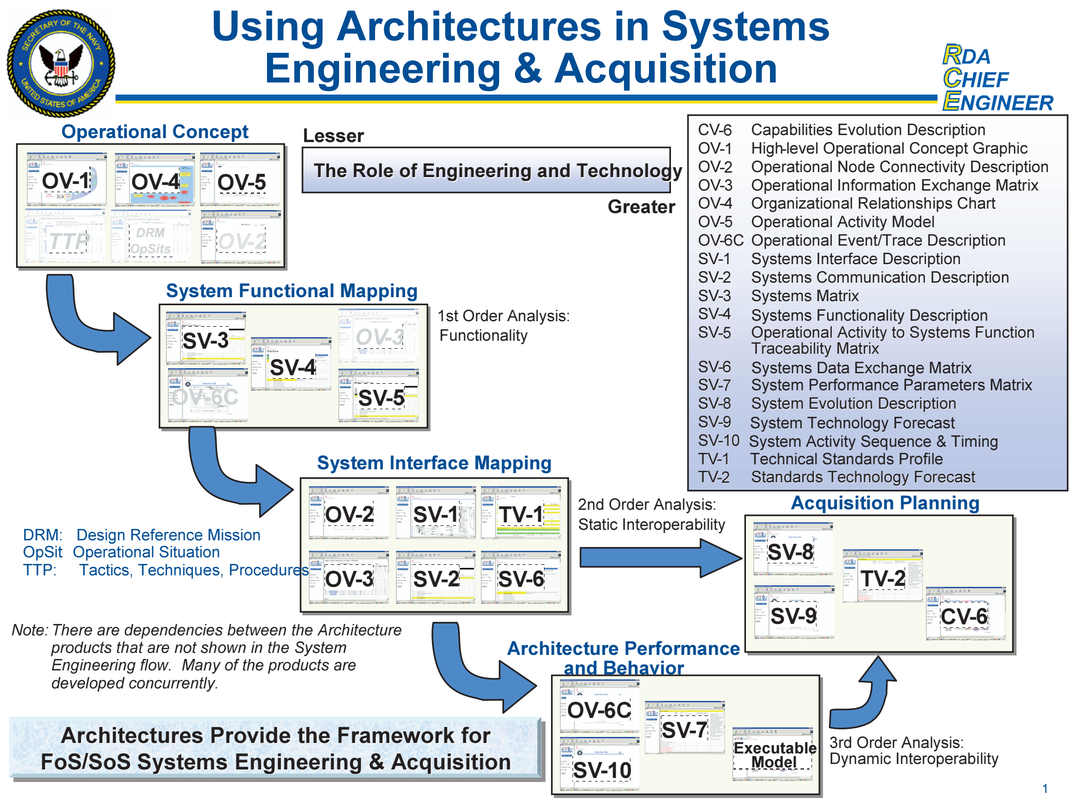


Figure 196: Using the Architecture Framework for Systems-of-Systems Systems Engineering and Acquisition [134].

D.0.4 What is the Difference Between a System-of-Systems and an Architecture?

The distinction between systems-of-systems and architectures is a subtle but important one that is best explained by analogy. The architecture diagram for a Canon PowerShot digital camera is shown in Figure 197. The function of the depicted architecture is to take pictures and either store them on a computer or print them for viewing. Necessary elements include the camera, a power source, a storage medium, an interface to the printing/storage device, and the printing/storage device. Selecting an item to perform each of these functions defines the *system-of-systems*. Choosing between the compact photo printer, the card photo printer, or a Bubble Jet printer results in a different system-of-systems within the same architecture. Similarly, replacing the PowerShot S2 IS (a 5 megapixel digital camera with a 12x optical zoom) with the new PowerShot S3 IS (identical except for a 6 megapixel resolution) produces a *new system-of-systems within the same architecture.*

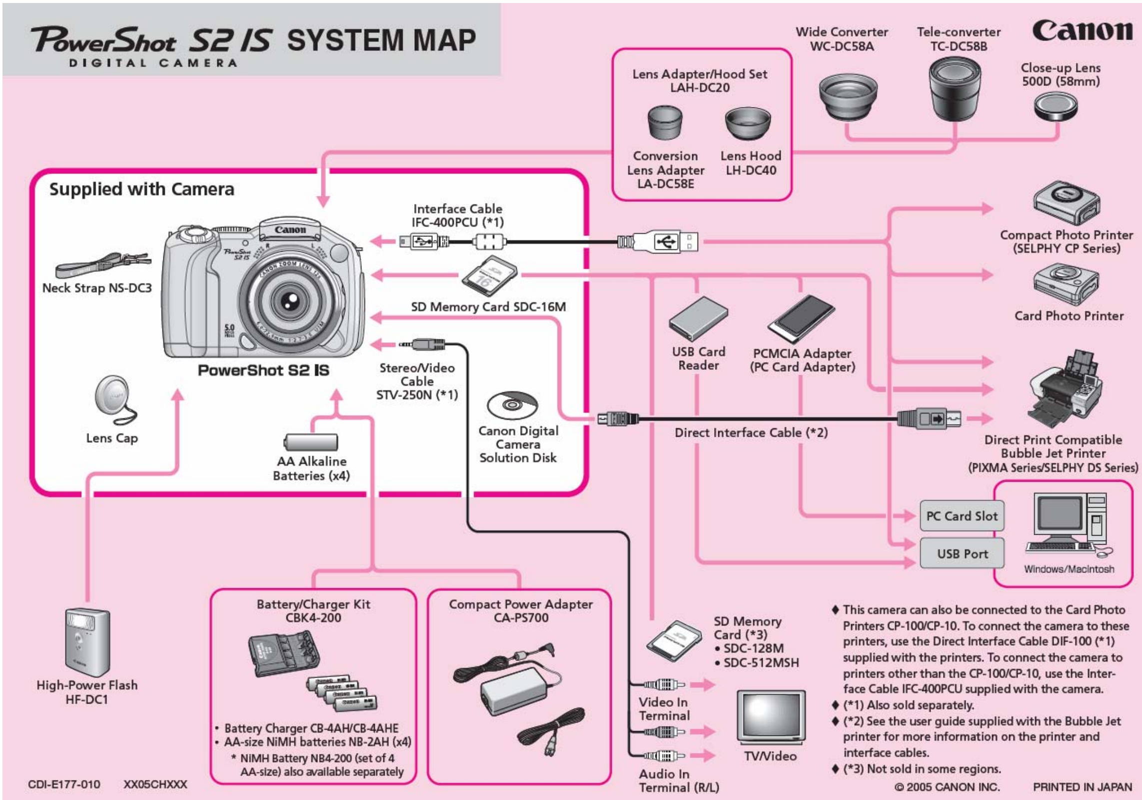


Figure 197: Architecture Diagram for a Canon PowerShot S2 IS Digital Camera [84].

An example of a different architecture that performs the same function is the use of a film camera instead of a digital camera. The power source is still a battery, albeit film cameras typically use fewer batteries than digital cameras. The storage medium would be film as opposed to memory cards. Most notably, the direct interface to the printing medium does not exist for film cameras: printing pictures requires a photo lab, a photo technician, developing chemicals, and a transportation system to deliver the film to the photo lab. The invention of the digital camera altered the traditional architecture for a mature function: taking pictures.

The above architectures operate within the framework of photography. The function performed by the photographic architecture is to capture an image; however, an image could also be captured within the framework of painting. A painting architecture would require a painter, paints, a canvas, and a subject to be depicted. This illustrative example shows how the same function can be performed in different frameworks, using different architectures, and with alternative components that define different systems-of-systems.

The camera example is illustrative of a heterogeneous system architecture: the function "capture an image" is performed using system elements that are functionally related but physically distinct[2]. For this research, **the focus is primarily on the application of technologies to different systems within a relatively static architecture**. This situation most closely resembles the motivating challenge of technology refresh for an existing military system architecture.

[2]In contrast, a homogeneous system architecture has elements which are also physically similar and have identical interfaces such as a brigade of M-1 tanks or an LGB model train set.

APPENDIX E

REVIEW OF PHYSICS-BASED MODELS

E.1 Physics-Based Model for Aircraft Flight

While many models are used within the FLAMES simulation to calculate mission effectiveness, the primary model created for this research is a performance model of a fixed wing aircraft. The code was written in C and compiled using Microsoft® Visual Studio 2003. This chapter summarizes the equations used in the development of the simple aircraft flight model and describes the equations used. All units used by the FLAMES kernel are in SI.

When assigned by the battle manager, an aircraft in the FLAMES scenario is commanded to fly at maximum speed to engage the target. To calculate the necessary forces on the aircraft, the current altitude of the aircraft is used to get the atmospheric density, ρ, from the 1976 Standard Atmosphere table. Then, the dynamic pressure, q, is calculated as:

$$q = \frac{1}{2}\rho V^2 \tag{33}$$

The weight of the platform is calculated as the sum of the empty weight, current payload weight, and the current weight of fuel in the aircraft. If the aircraft needs to turn to reach its intended position, the flight model receives as an input the commanded turn G. A check is performed to see if the commanded turn G and commanded roll rate exceed the limits of the platform as defined by the user prior to execution. In addition to these limits, the turn can be constrained by the lift available from the aircraft. The maximum turn acceleration of the aircraft due to lift, n_{MaxCL}, is therefore calculated as:

$$n_{MaxCL} = \frac{C_L q S}{W} \tag{34}$$

Where C_L is the maximum lift coefficient of the aircraft in its current configuration, q is the dynamic pressure, S is the user defined wing area, and W is the instantaneous weight of the aircraft. The load factor in a turn is therefore the minimum of the lift-limited turn and the commanded turn G of the aircraft. From the load factor, the turn radius, R, and turn rate, ϕ, can be calculated as:

$$R = \frac{V^2}{n} \tag{35}$$

$$\phi = \frac{V}{R} \tag{36}$$

Next, the drag on the aircraft, D, is calculated as a function of dynamic pressure, drag coefficient, and wing area according to Equation 37

$$D = C_D q S \tag{37}$$

Where C_D is defined by the user as an element of the aircraft DOE. Next, since the battle manager commands the aircraft to fly at maximum speed, a calculation is needed to see if this speed is achievable with the thrust and drag of the aircraft at the current flight altitude and speed. The desired axial acceleration of the aircraft to reach the commanded speed in timestep Δt is:

$$a = \frac{(V_{commanded} - V_{current})}{\Delta t} \tag{38}$$

The thrust required, T, to achieve this acceleration is therefore:

$$T = a\left(\frac{W}{g}\right) + D + W\cos\theta \tag{39}$$

Where W is the aircraft weight and θ is the angle between the aircraft's flight direction and the velocity vector. If the thrust available from the aircraft's engine exceeds the thrust required, then the thrust required is the actual thrust and the aircraft accelerates toward the commanded speed at this time step. On the other hand, if the thrust required is greater than the thrust available then the aircraft is not be able to accelerate to the commanded speed until some fuel is burned off or the altitude is increased to lower the magnitude of drag.

The speed at the next time step is then calculated by adding the resultant axial acceleration to the current velocity to get the speed at the next time step. The position of the aircraft is also updated to incorporate the velocity at the current time step and the orientation of the aircraft is updated to take into account the roll angle if a turn is required.

Next, to calculate the new weight of the aircraft, the fuel is decremented by the amount of fuel burned at this time step using Equation 40.

$$\Delta Fuel = -TSFC \times T \times \Delta t \qquad (40)$$

If the fuel falls below zero at any point in the mission, the aircraft is destroyed.

These equations are repeated for each aircraft at every time step. The time step value is a user defined parameter in the FLAMES scenario. Experimental observation indicated that a parameter on the order of 1-2 seconds is appropriate to balance run time and fidelity. While aircraft cruise is relatively insensitive to large time steps, aircraft and missile turn calculations produce inaccurate results for time steps greater than five seconds.

E.2 Calibration of IADS Model

To support identification of technologies across a range of scenarios within the Iraq scenario, it is necessary to establish a parametric threat model that represents the enemy IADS. Characteristics of the KARI IADS system are summarized in Section 5.2.1.5. The Gulf War Air Power survey provides a detailed recreation of the opening events of Operation Desert Storm [105]. Historical references also identify the F-117A as the only aircraft that bombed targets in downtown Baghdad on the opening night of the war. Reviewing this account identifies likely gaps in the IADS radar coverage through which the F-117A was able to maneuver. Using the FLAMES sensor coverage package, it is possible to calibrate the input values for the Iraqi radars to develop a coverage map that provides access corridors for F-117 aircraft and denies access to non-stealthy platforms such as the B-52 [399]. The sensor coverage window used to calibrate the hostile radars is shown in Figure 198.

The sensor coverage is a function of altitude, which was defined as 12,000 m (39,370 ft). The region for which to calculate coverage was a polygon defined around the borders of Iraq. All units with a "Red Ground Radar," the primary sensor used by hostile units, were added to the coverage list. Two options for radar coverage are available: radial and region. "Radial coverage can be used to provide a very clear picture of the coverage of a single sensor or of several sensors where each one's coverage area does not overlap" but is not as useful for overlapping sensors [399]. For this reason, region coverage was used. Regions are color coded areas that depict overlapping radars and are classified by the minimum number of detections in that region. Four regions with detection levels of 1, 2, 5, and 10 minimum detections were defined as green, blue, orange, and red respectively. As the color of a region moves towards red, the probability of the selected platform being detected inside that region increases. The purpose of the calibration exercise is to calculate the required transmission power of the radar to match the the computed coverage area to the approximate coverage areas discerned from the Gulf War Air Power Survey [105].

In addition to the unknown transmission power, the simple radar model created for FLAMES based on the radar range equation requires a user defined frequency input, signal bandwidth (MHz), noise figure, and signal-to-noise ratio which were set to 3000 MHz, 1.0 MHz, 3.0, and 1.0 dB respectively. If the frequency $\pm$ the signal bandwidth does not encompass the frequency of the target's RCS, it is not possible to detect the target with the given radar. The other two factors were set to baseline values from the example FLAMES ground radar model. The minimum detectable signal (MDS) is defined as:

$$MDS = kT_sBN_fSNR \tag{41}$$

Where k is Boltmann's constant, T_s is the standard temperature in Kelvin, B is the signal bandwidth (MHz), N_f is the noise figure, and SNR is the signal-to-noise ratio. A detection occurs when the received power of the radar is greater than the MDS, neglecting jamming. The received power is calculated as:

$$P_R = \frac{P_T G_T G_R \lambda^2 \sigma}{(4\pi)^3 R^4} \tag{42}$$

Where P_T is the power of the transmitter, G_T is the gain of the transmitter, G_R is the gain of the receiver, λ is the wavelength of the transmitted signal, σ is the radar cross section of the target, and R is the distance from the radar to the target. The gain values were fixed for the antenna used by the "Red Ground Sensor." After setting the radar cross section for the F-117A and B-52H at the values noted by Stonier [386], an iterative process was used to determine the baseline transmission power. This value was equal to 15 dBW. Using this value, the sensor coverage diagrams for the B-52H and F-117A are shown in Figures

199 and 200 respectively, which graphically shows the difference between non-stealthy and stealthy platforms in terms of detectability. The percentage of Iraq covered by 1, 2, 5, and 10 minimum detections is summarized in Table 34. Based on this table, the B-52H can be detected at least once over 61.94% of Iraq, which the F-117A's stealth permits at least one detection over about 33.03% of the country.

Table 34: Comparison of Coverage Regions for B-52H (non-stealthy) and F-117A (stealthy) Platforms.

Min.	Platform	
Detects	**B-52H**	**F-117A**
1	15.97%	21.08%
2	26.70%	8.85%
5	13.21%	2.08%
10	6.06%	1.02%
Total	61.94%	33.03%

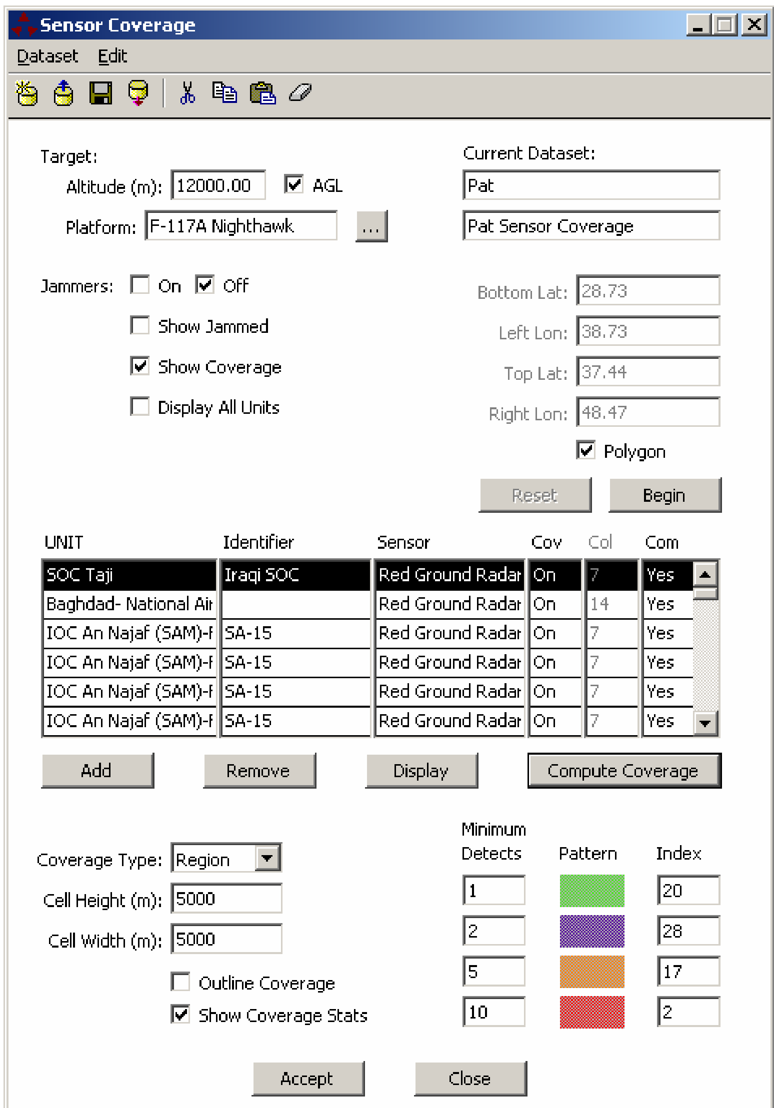


Figure 198: Defining the Input Parameters for the Sensor Coverage Calculations.

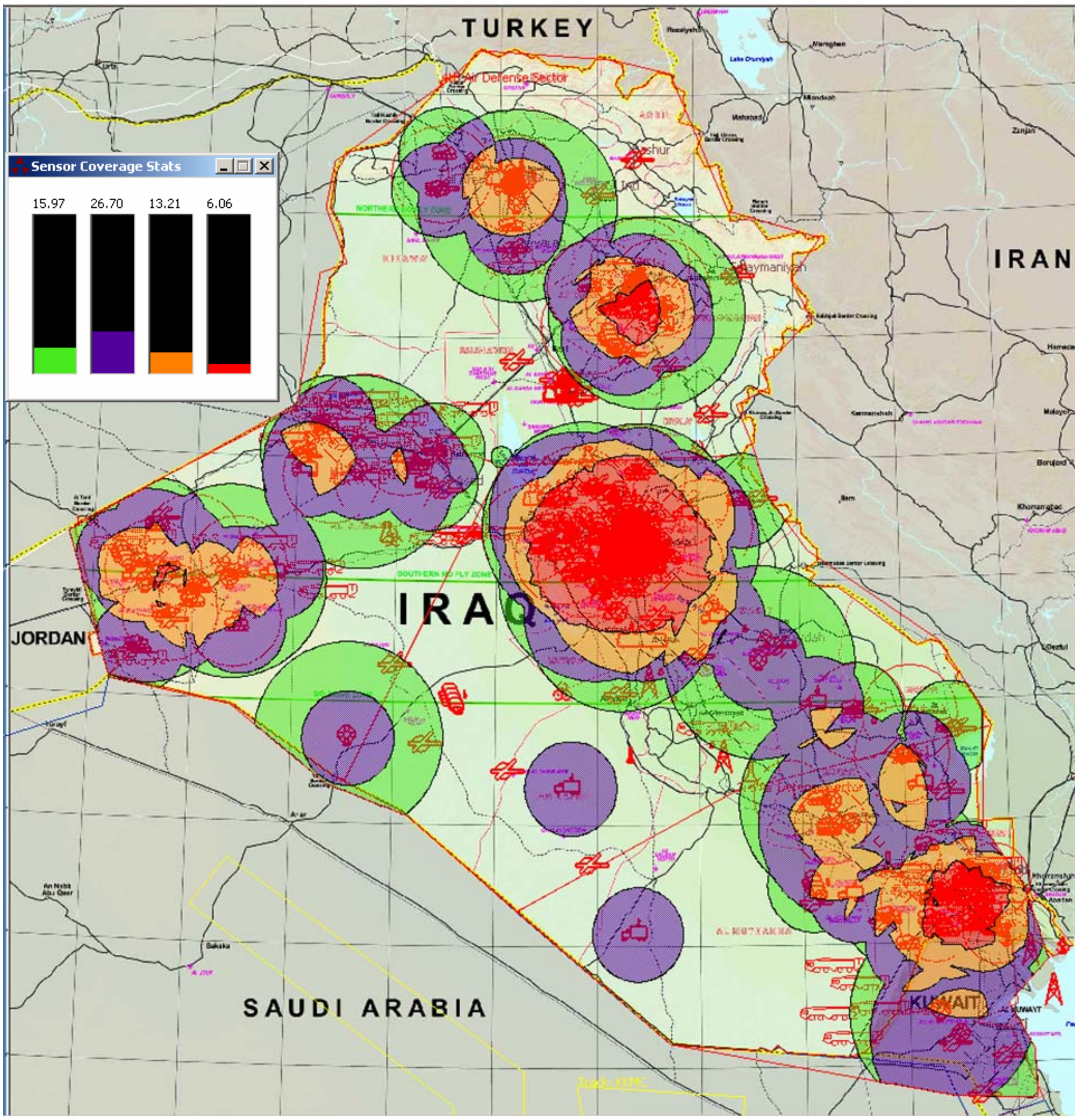


Figure 199: IADS Coverage Map of Iraq: B-52H Platform at 12,000 m.

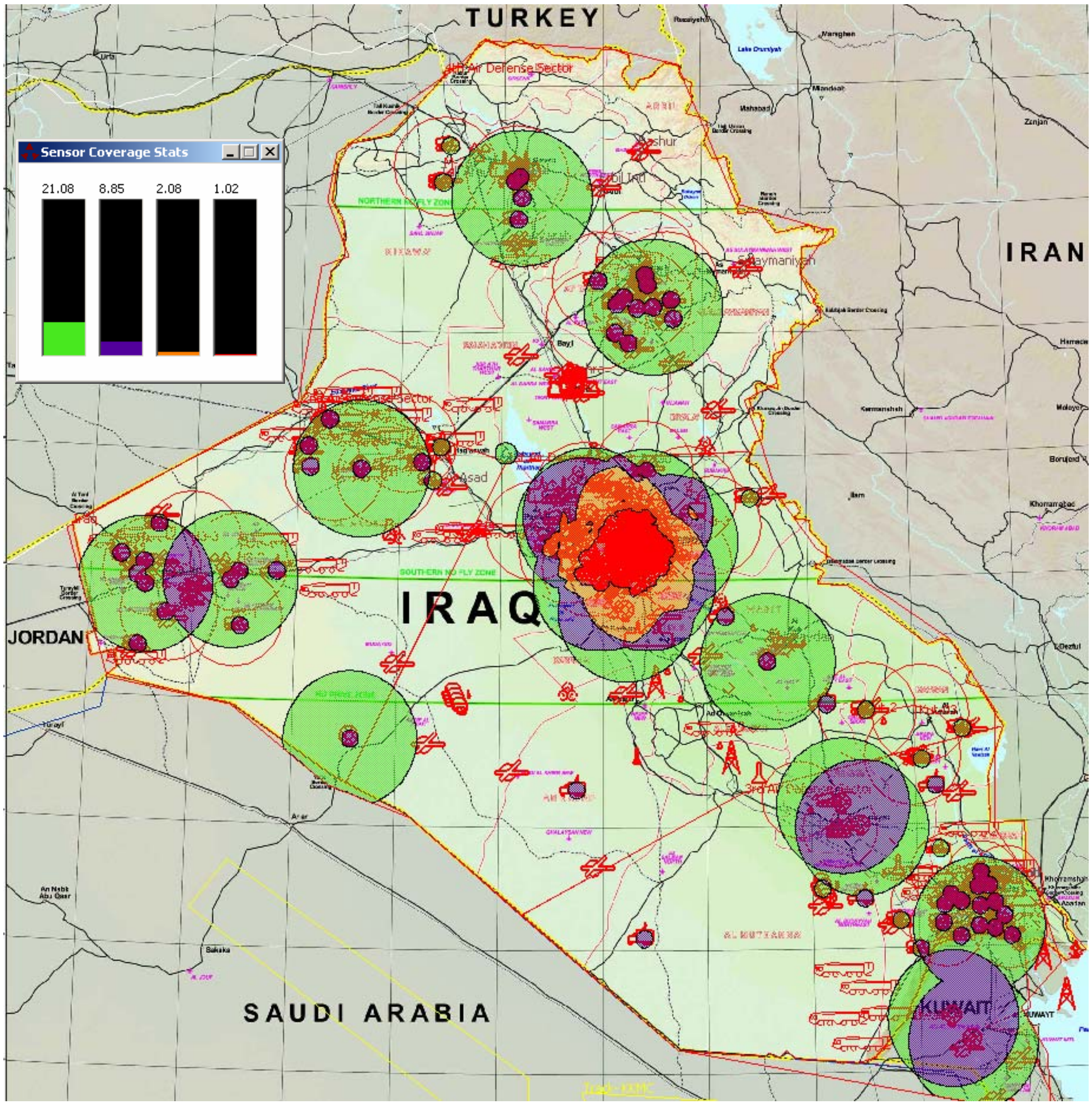


Figure 200: IADS Coverage Map of Iraq: F-117A Platform at 12,000 m.

REFERENCES

[1] "20Q, The Neural Net on the Internet." Online at www.20q.net. Accessed March 17, 2006.

[2] "Aerospaceweb.org." Online at http://www.aerospaceweb.org/. Accessed May 5, 2006.

[3] "Agents for Net-Centric Warfare and Time Critical Targets." White Paper, Computer Technology Associates, Inc., Online at www.cta.com/content/docs/casestudies/netcentric_tct.pdf. Accessed June 12, 2006.

[4] "ARTiSAN Software Home Page." Online at www.artisansw.com. Last Accessed October 5, 2005.

[5] "B-1 Lancer." Online at http://en.wikipedia.org/wiki/B-1_lancer. Last Accessed July 25, 2006.

[6] "Breakaway Federal Systems Home Page." Online at http://www.breakawayfederal.com/. Last Accessed January 13, 2007.

[7] "Briefing, 28th Air Expeditionary Wing, Subject: Operation Enduring Freedom," December 17, 2001. Quoted in Theisen, E. E. *Ground Aided Precision Strike, Heavy Bomber Activity in Operation Enduring Freedom*, Air War College Maxwell Paper No. 31, Air University Press, Maxwell Air Force Base, Alabama, 2003.

[8] "Comparison of Fighter Aircraft Combat Radius." Online at http://www.f-16.net/f-16_forum_viewtopic-t-2242.html. Accessed May 15, 2006.

[9] "Definition: Simulation." Online at http://www.webopedia.com/TERM/S/simulation.html. Last Accessed October 12, 2005.

[10] "F/A-18 Super Hornet Fact Sheet." Online at http://www.zap16.com/mil Last Accessed March 12, 2006.

[11] "IBM to Acquire Rational Software." Online at http://www.ibm.com/news/us/2002/12/061.html. Last Accessed November 15, 2005.

[12] "Systems Modeling Language Forum." Online at http://www.sysml.org. Accessed September 2005.

[13] "The Air Force Research Lab: July 2005 Accomplishments." Online at http://www.afrl.af.mil/accomprpt/jul05/accompjul05.asp. Last Accessed October 2005.

[14] "Tomahawk SLCM- Ship Launched Cruise Missile." Online at http://navysite.de/weapons/tomahawk.htm. Last Accessed November 13, 2005.

[15] "Venik's Aviation Page." Online at http://www.aeronautics.ru/. Last Accessed November 7, 2005.

[16] "Wikipedia: The Free Online Encyclopedia." www.wikipedia.org.

[17] "DARPA Neural Network Study," tech. rep., M.I.T. Lincoln Laboratory, Lexington, MA, 1988.

[18] "JMP 5.1.2 Help File." Computer Program, SAS Institute, Inc. Cary, NC, 1989-2005.

[19] "JMP, Version 6." Computer Program, SAS Institute, Inc. Cary, NC, 1989-2005.

[20] "New World Vistas: Air and Space Power for the 21st Century," tech. rep., United States Air Force Scientific Advisory Board, 1996.

[21] "White Paper on Long Range Bombers," tech. rep., U.S. Air Force, 1999.

[22] *The American Heritage Dictionary of the English Language, Fourth Edition.* Houghton Mifflin Company, 2000.

[23] "Network Centric Warfare," tech. rep., Department of Defense Report to Congress, July 2001.

[24] "U.S. Air Force Long-Range Strike Aircraft White Paper," tech. rep., U.S. Air Force, 2001.

[25] "Multi-Agent Learning: Theory and Practice." Workshop at the Neural Information Processing Systems Conference, Vancouver, Canada, December 13-14, 2002.

[26] "Visual Studio .NET 2003." Computer Program, Microsoft Corporation, Redmond, WA, 2003.

[27] *Defense Acquisition Guidebook.* U.S. Department of Defense, 2004.

[28] "FLexible Analysis Modeling and Exercise System (FLAMES)." Computer Program, Ternion, Inc., Huntsville, AL, December 2004.

[29] "U.S. Air Force Policy Letter Digest." Online at http://www.af.mil/library/policy/letters/pl2004_04.html, April 2004. Last Accessed December 2005.

[30] "ModelCenter Version 6.1." Computer Program, Phoenix Integration, Blacksburg, VA, 2005.

[31] "Rational Software Modeler." Computer Program, IBM, Armonk, New York, 2005.

[32] "Telelogic Acquires US Based Leader of Enterprise Architecture Tools." Telelogic Press Releases, Online at http://www.telelogic.com/news/newsroom/releases.cfm, April 2005.

[33] "Google Earth." Computer Program, Google, Inc., 2005-2007.

[34] "Technology For Affordability: A Report on the Activities of the Working Groups - Integrated Product/Process Development (IPPD), Simplified Contracting, Dual-Use Manufacturing," tech. rep., National Center for Advanced Technologies, December 1993.

[35] "Simulation-Based Engineering Science: Revolutionizing Engineering Science Through Simulation," tech. rep., National Science Foundation, February 2006.

[36] "Report on System-of-Systems Engineering for Air Force Capability Development, Executive Summary and Annotated Brief," tech. rep., SAB-TR-05-04, United States Air Force Scientific Advisory Board, July 2005.

[37] "Alternatives for Long-Range Ground-Attack Systems," tech. rep., Congressional Budget Office, March 2006.

[38] *Integrated Theater Engagement Model (ITEM) Technical Manual, Version 8.3.* Science Applications International Corporation, 10260 Campus Point Drive, San Diego, California 92121, November 5, 1999.

[39] Acquisition Reform Office, ASN (RD&A), Department Of the Navy, "Work Book for Video Series on Integrated Product And Process Development." National Center for Advanced Technologies, 1997.

[40] ACSYNT Institute, "ACSYNT Overview and Installation Manual," 1992.

[41] Alberts, D. S. and Hayes, R. E., *Understanding Command and Control*. Department of Defense Command and Control Research Program, 2006.

[42] American Institute of Aeronautics and Astronautics, "Guide for the Preparation of Operational Concept Documents." ANSI/AIAA G-043-200x, Draft 2.0, August 26 2006. Draft AIAA Standard.

[43] American Society for Quality Control, "Glossary & Tables for Statistical Quality Control." http://www.itl.nist.gov/div898/handbook/pri/section3/pri334.htm, 1983.

[44] Anderson, D. J., Campbell, J. E., and Chapman, L. D., "Evaluating a Complex System of Systems Using State Modeling and Simulation." Powerpoint Presentation Online at www.dtic.mil/ndia/2003systems/abst.ppt, October 2003.

[45] Arnold, S., "Placing Defense Capability in the Systems Engineering Lexicon," *INCOSE Insight*, vol. 8, no. 1, pp. 9–11, October 2005.

[46] Aubin, S. P., "Newsweek and the 14 Tanks," *Air Force Magazine*, July 2000.

[47] Australian Government Department of Defense, "Definition: Capability." www.defence.gov.au/budget/03-04/dar/07_18_glossary.htm, Last Accessed October 15, 2005.

[48] Avigation Networks, Inc., "World Aeronautical Database: Airports." Online at http://worldaerodata.com/, Last Accessed March 15, 2006.

[49] Baker, A. P., *The Role of Mission Requirements, Vehicle Attributes, Technologies and Uncertainty in Rotorcraft System Design*. PhD thesis, Georgia Institute of Technology, 2002.

[50] Baker, A. P., Mavris, D. N., "Assessing the Simultaneous Impact of Requirements, Vehicle Characteristics and Technologies During Aircraft Design." AIAA-01-0533, Presented at the 39th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 8-11, 2001.

[51] Ball, R. E., *The Fundamentals of Aircraft Combat Survivability Analysis and Design*. American Institute of Aeronautics and Astronautics, 2003.

[52] Bandte, O., *A Probabilistic Multi-Criteria Decision Making Technique for Conceptual and Preliminary Aerospace Systems Design*. PhD thesis, Georgia Institute of Technology, 2001.

[53] Bandte, O., Mavris, D.N., DeLaurentis, D.A., "Viable Designs Through a Joint Probabilistic Estimation Technique." Presented at the AIAA/SAE World Aviation Congress, October 1999.

[54] Bar-Yam, Y., *Dynamics of Complex Systems*. Addison-Wesley, 1997.

[55] Barros, P. A., Kirby, M. R., and Mavris, D. N., "Impact of Sampling Technique Selection on the Creation of Response Surface Models." Presented at the 2004 SAE World Aviation Congress, AIAA 2004-01-3134, 2004.

[56] Berry, B. J. L., "Cities as Systems Within Systems of Cities," *Papers of Regional Sciences Association*, vol. 13, pp. 147–163, 1964.

[57] Bert, C. W., "Prediction of Range and Endurance of Jet Aircraft at Constant Altitude," *Journal of Aircraft*, no. 18, pp. 890–892, 1981.

[58] Betz, F., *Managing Technological Innovation: Competitive Advantage from Change*. John Wiley and Sons, Inc., 1998.

[59] Bille, M. and Lorenz, R., "Requirements for a Conventional Prompt Global Strike Capability." Presented at the NDIA Missile and Rockets Symposium and Exhibition, Online at www.dtic.mil/ndia/2001missiles/bille.pdf, May 2001.

[60] Biltgen, P. T., "Concept Identification and Selection Methods for a Solid-Propellant Target Vehicle to Support the Development of a National Missile Defense (NMD) System," tech. rep., AE 8900 Special Topics Report, Georgia Institute of Technology, July 29, 2004.

[61] Biltgen, P. T. and Ender, T. R., "Demonstration of a Collaborative Capability-Based Design Environment." Powerpoint Presentation at the 13th Annual Aerospace Systems Design Laboratory External Advisory Board Review, May 4, 2005.

[62] Biltgen, P. T., Ender, T. R., and Mavris, D. N., "Development of a Collaborative Capability-Based Tradeoff Environment for Complex System Architectures." AIAA 2006-0728, Presented at the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 9-12, 2006.

[63] Biltgen, P. T., et al., "Proteus: A Long Range Liquid Booster Target Vehicle," tech. rep., AIAA Missile Systems Technical Committee Graduate Strategic Missile Design Competition, Final Report, 2004.

[64] Blackhurst, J., "New Directions in Partnering." Powerpoint Presentation, Online at http://www.oai.org/Event/presentation/afrl.ppt, November 16, 2005.

[65] Bocetti, C. I., Bart, J. R., Kepler, C. B., Sykes, P. W., Probst, J. R., "Development of a Spatially-Explicit, Individual-Based Model to Simulate Kirtlands Warbler Population Dynamics." USGS Patuxent Wildlife Research Center, http://www.pwrc.usgs.gov/research/sis98/bocett1s.htm, 2004.

[66] Bolkcom, C., "Military Suppression of Enemy Air Defenses (SEAD): Assessing Future Needs," tech. rep., Congressional Research Service Report for Congress, Report RS21141, May 11, 2005.

[67] Bowers, J., "AFIT Students Gain Operational Insight During Class." Published by the Air Force Research Laboratory, Wright-Patterson AFB, Ohio, Online at http://www.afrl.af.mil/articles/081406_AFIT_students.asp, Last Accessed August 2006.

[68] Bowie, C. J., "Destroying Mobile Ground Targets in an Anti-Access Environment," tech. rep., Northrop Grumman Analysis Center Papers, December 2001.

[69] Bowling, M. and Veloso, M. M., "Multiagent Learning Using a Variable Learning Rate," *Artificial Intelligence*, no. 136, pp. 215–250, 2002.

[70] Box, G. E. P. and Wilson, K. B., "On the Experiemental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society*, vol. B13, pp. 1–38, 1951.

[71] Box, G. E. P., Hunter, J. S., and Hunter, W. G., *Statistics for Experimenters: Design, Innovation and Discovery, 2nd Edition*. John Wiley and Sons, New York, 2005.

[72] Braun, R.D. and Kroo, I.M., *Collaborative Optimization: An Architecture for Large-Scale Distributed Design*. PhD thesis, Stanford University, June 1996.

[73] Breguet, L., "Calcul du poids de combustible consummé par un avion en vol ascendant," *Comptes Rendus de l'academie des sciences*, no. 177, pp. 870–872, 1923.

[74] Brodie, B., *The Absolute Weapon: Atomic Power and World Order*. Ayer Company Publishers, 1946.

[75] Brown, D., "Comments to the Author on Technology Forecasting." Atlanta, GA, February 21, 2007.

[76] Brown, D. A., "Quantitative Technology Assessment in AFRL." Presented at the Department of Defense M&S Conference, May 4, 2006.

[77] Brown, David. Personal Communication with the Author, January 10, 2007. Used with Permission.

[78] Brown, David. Personal Interview, Atlanta, Georgia, July 13, 2006.

[79] Buonanno M. and Mavris, D., "A New Method for Aircraft Concept Selection Using Multicriteria Interactive Genetic Algorithms, AIAA-2005-1020." Presented at the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan. 10-13, 2005.

[80] Bush, G. W., "The National Security Strategy of the United States of America." Online at http://www.whitehouse.gov/nsc/nss.html, September 2002.

[81] Bush, G. W., "President Bush Addresses the Nation." Transcript online at http://www.whitehouse.gov/news/releases/2003/03/20030319-17.html, March 19 2003.

[82] Calise, A. J., "Research in Neural Network Based Robust Adaptive Control with Application to Alternate Control Technology Missile Autopilot Design," tech. rep., Georgia Institute of Technology, School of Aerospace Engineering, 1995-1997.

[83] Canemaker, John, *Paper Dreams: The Art and Artists of Disney Storyboards.* Disney Editions, 1999.

[84] Canon Digital Imaging, "PowerShot S2 IS System Map." Online at http://www.usa.canon.com/consumer/, Last Accessed July 2006.

[85] Carlone, R. V., Blair, M., Obenski, S., and Bridickas, P., "Patriot Missile Software Problem." GAO/IMTEC-92-26, United States General Accounting Office Information Management and Technology Division, February 4, 1992.

[86] Carlson, B., "Enabling Warfighting Transformation." Presented at MILCOM 2005, October 19, 2005.

[87] Carter, C. E., Coker, P. D., and Gorenc, S., "Dynamic Commitment: Wargaming Projected Forces Against the QDR Defense Strategy," *National Defense University Institute for Strategic Studies Strategic Forum*, no. 131, November 1997.

[88] Casal, A. and Hogg, T., "Design Principles for the Distributed Control of Modular Self-Reconfigurable Robots," in *Collectives and the Design of Complex Systems*, 2004.

[89] Casti, J. L., *Would-Be Worlds.* John Wiley and Sons, 1996.

[90] Caudill, D., Zeh, J., et. al., "Aerospace Vehicle Technology Assessment & Simulation (AVTAS) Mission Level Simulation System (MLS2)." AIAA 2004-4944, Presented at the AIAA Modeling and Simulation Technologies Conference and Exhibit, Providence, Rhode Island, August 16-19, 2004.

[91] Central Intelligence Agency, "World Factbook 2006." Online at http://www.cia.gov/cia/publications/factbook/, Last Updated June 1, 2006.

[92] Central Intelligence Agency, "Iraq's Weapons of Mass Destruction Programs." Online at http://www.cia.gov/cia/reports/iraq_wmd/Iraq_Oct_2002.htm, October 2002.

[93] Central Intelligence Agency, "Iraq Country Profile." Map Number 764225, Online at http://www.cia.gov/cia/publications/mapspub/80.shtml, Last Accessed June 5, 2006., January 2003.

[94] Christensen, C. M., *The Innovator's Dilemma: The Revolutionary Book That Will Change the Way You Do Business.* Collins Business Essentials, 1997.

[95] Christie, R., "U.S. Air Force to Step Up New Bomber Search in Next Budget." Dow Jones MarketWatch, Online at http://www.globalsecurity.org/org/news/2006/060629-usaf-bomber.htm, Last Accessed June 29, 2006.

[96] Cioppa, T. M., *Efficient Nearly Orthogonal and Space-Filling Experimental Designs for High-Dimensional Complex Models*. PhD thesis, Naval Postgraduate School, Monterey, California, 2002.

[97] Clancy, T., *Carrier: A Guided Tour of an Aircraft Carrier*. Berkley Books, New York, 1999.

[98] Clancy, T., *Every Man a Tiger*. Berkley Books, New York, 2005.

[99] Clancy, T., *Submarine: A Guided Tour Inside a Nuclear Warship*. Berkley Books, New York, January 2002.

[100] Clausewitz, C. V., *On War (Vom Kriege)*. Dummlers Verlag, Berlin 1832, First Edition Translation Published online by Project Gutenberg, 1999.

[101] Clough, B. T., "Air Force Research Laboratory Air Vehicles Directorate: Capability-Focused Tech Investment." Powerpoint Presentation, Approved for Public Release, December 2005.

[102] Cochran, W. G., and Cox, G. M., *Experimental Designs*. Wiley, 1950.

[103] Coffin, J. G., "A Study of Airplane Range and Useful Loads." NACA Report 69, 1919.

[104] Cohen, E. A., "Gulf War Air Power Survey, Volume I: Planning and Command and Control," tech. rep., Washington, D. C., 1993.

[105] Cohen, E. A., "Gulf War Air Power Survey, Volume II: Operations and Effects and Effectiveness," tech. rep., Washington, D. C., 1993.

[106] Cohen, E. A., "Gulf War Air Power Survey, Volume V: A Statistical Compendium and Chronology," tech. rep., Washington, D. C., 1993.

[107] Cordesman, A. H., "Key Targets in Iraq." Center for Strategic and International Studies, Washington, D.C., February 1998.

[108] Cortes, L., "B-2 Drops Pair of 4,700-Pound GBU-37 GAMs on Iraqi Communications Tower," *Defense Daily*, March 31, 2003.

[109] Creighton, L., Jones, B., Sall, J., and Zangi, A., "The JMP Advantage." White Paper, Online at http://www.jmp.com/software/whitepapers/, 2005. Last Accessed December 2, 2006.

[110] Crisp, H. and Ewald, B., "Capability Engineering for Systems of Systems: A Coalition Perspective," *INCOSE INSIGHT*, vol. 8, no. 1, pp. 3, 7, October 2005.

[111] Crossley, W. A., and Mane, M., "System of Systems Inspired Aircraft Sizing Applied to Commercial Aircraft / Airline Problems." AIAA 2005-7426, Presented at the AIAA 5th Aviation, Technology, Integration, and Operations Conference (ATIO), Arlington, Virginia, September 26-28, 2005.

[112] Crowder, G., "Effects-Based Operations." Powerpoint Briefing.

[113] Cruse, T. A., "Non-Deterministic Non-Traditional Methods (NDNTM)." NASA/CR2001-210976, July 2001.

[114] Cruse, T. A., "Comments to the Air Force Integrated Collaborative Environment (AF-ICE) Industry Team," December 20, 2005.

[115] Culver, W., "Distributed Wargaming Simulation." Presented at the 2005 FLAMES User Group Conference, June 2005.

[116] Daberkow, D. D. and Mavris, D. N., "New Approaches to Conceptual and Preliminary Aircraft Design: A Comparative Assessment of a Neural Network Formulation and a Response Surface Methodology." Presented at the 3rd World Aviation Congress and Exposition, Anaheim, CA, September 28-30, 1998.

[117] Danner, T. W., *A Formulation of Multidimensional Growth Models for the Assessment and Forecast of Technology Attributes.* PhD thesis, Georgia Institute of Technology, 2006.

[118] Davis, R. G., *Decisive Force: Strategic Bombing in the Gulf War.* Air Force Historical Studies Office, GPO Stock No.008-070-00710-0, 1996.

[119] Davis, P. K., *Effects-Based Operations: A Grand Challenge for the Analytical Community.* RAND Corporation, 2001.

[120] Davis, P. K., *Analytic Architecture for Capabilities-Based Planning, Mission-System Analysis, and Transformation.* RAND National Defense Research Institute, MR-1513-OSD, 2002.

[121] Daw, A. J., "New Process and Structure Thinking for Capability Development." Presented at the 9th International Command and Control Research and Technology Symposium, Copenhagen, Denmark, Sept 14-16, 2004.

[122] Deb, K., "Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.

[123] Defense Acquisition University, "Background of the Joint Capabilities Integration and Development System (JCIDS)." PM-2001-ISE, Online at https://acc.dau.mil/CommunityBrowser.aspx?id=18467.

[124] Defense Acquisition University, "Integrated Defense Acquisition, Technology and Logistics Life Cycle Management Framework Chart." Online at http://akss.dau.mil/ifc/.

[125] Defense Advanced Research Projects Agency, "DARPA Mission and Overview." Online at http://www.darpa.mil/body/mission.html, Last Accessed March 2006.

[126] Delaney, P. J., "What is Simulation? (Another View)." https://www.amso.army.mil/library/primers/what-is, Accessed August 20, 2005.

[127] DeLaurentis, D., Lim, C., Kang, T., Mavris, D.N., Schrage, D., "System-of-Systems Modeling for Personal Air Vehicles." Presented at the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, September 4-6, 2002.

[128] Demuth, H. and Beale M., "MATLAB: Neural Network Toolbox User's Guide Version 4," The MathWorks, Inc., 2004.

[129] Department of Defense, "Rotary Wing Vehicle Technology Development Approach (TDA) 4.0, Technology Efforts and Objectives (TEO)," 1994.

[130] Department of Defense Systems Management College, ed., *Systems Engineering Fundamentals*. Defense Acquisition University Press, January 2001.

[131] Department of the Air Force, "Fiscal Year (FY) 2006/2007 Budged Estimates: Research, Development, Test and Evaluation (RDT&E) Descriptive Summaries," vol. II, p. 802.

[132] Department of the Air Force, "Air Force Handbook- 109th Congress." Online at http://www.af.mil/library/posture/2005handbook.pdf, 2005.

[133] DeStefano, G. V., "Agent Based Simulation SEAS Evaluation of DoDAF Architecture," Master's thesis, Air Force Institute of Technology, 2004.

[134] Dickerson, C. E., Soules, S. M., Sabins, M. R., and Charles, P. H., *Using Architectures For Research, Development, and Acquisition.*

[135] Dieter, G. F., *Engineering Design: A Materials and Processing Approach*. McGraw-Hill, 1999.

[136] DoD Architecture Framework Working Group, "DoD Architecture Framework Version 1.0, Volume I: Definitions and Guidelines," February 9, 2004.

[137] DoD Architecture Framework Working Group, "DoD Architecture Framework Version 1.0, Volume II: Product Descriptions," February 9, 2004.

[138] Dupuy, T. N., Johnson, C., Bongard, D. L., and Dupuy, A. C., *How to Defeat Saddam Hussein*. Warner Books, 1991.

[139] Duquette, M., Nalepka, J. P., and Luczak, R., "The Enhanced Generic Air Defense System." AIAA 2004-4799, Presented at the AIAA Modeling and Simulation Technologies Conference and Exhibit, Providence, Rhode Island, August 2004.

[140] Eisenhower, D. D., "Annual Message to Congress on the State of the Union," January 9, 1959.

[141] Ender, T. R., *A System-of-Systems Approach to the Design of an Air-Defense Weapon*. PhD thesis, Georgia Institute of Technology, 2006.

[142] Ender, T.R., McClure, E.K., Mavris, D.N., "A Probabilistic Approach to the Conceptual Design of a Ship-Launched High Speed Standoff Missile." Presented at the AIAA 2002 Missile Sciences Conference, Monterey, CA, November 5-7, 2002.

[143] ENGLER, W., "Creation of a Set of Parametric Engine Models Utilizing Neural Networks in a Systems-of-Systems Context." AE8900 Special Topics Report, Georgia Institute of Technology, School of Aerospace Engineering, December 2005.

[144] ENGLER, W. O., BILTGEN, P. T. AND MAVRIS, D. N., "Concept Selection Using an Interactive Reconfigurable Matrix of Alternatives (IRMA)." Presented at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2007.

[145] ERIKSSON, H. E., PENKER, M., LYONS, B. AND FADO, D., *UML 2 Toolkit*. Wiley Publishing, Inc., 2004.

[146] EWING, C. M., "MS&A of Advanced Weapon Concepts in a System-of-Systems Environment Using FLAMES." Presentation at the 2003 FLAMES User Group Conference, Huntsville, AL, June 2003. Approved For Public Release AAC/PA-03-008.

[147] FANG, K. T., AND WANG, Y., *Number-Theoretic Methods in Statistics*. Chapman and Hall, London, 1994.

[148] FEDERATION OF AMERICAN SCIENTISTS, "B-52 Stratofortress." http://www.fas.org/nuke/guide/usa/bomber/b-52.htm, Last Accessed August 7, 2005.

[149] FEDERATION OF AMERICAN SCIENTISTS, "BGM-109 Tomahawk." http://www.fas.org/man/dod-101/sys/smart/bgm-109.htm, Last Accessed August 7, 2005.

[150] FEDERATION OF AMERICAN SCIENTISTS, "Digital Terrain Elevation Data [DTED]." Online at http://www.fas.org/irp/program/core/dted.htm, Last Accessed March 3, 2006.

[151] FERGUSON, T. S., *Game Theory*. Online Textbook, University of California at Los Angeles.

[152] FISHER, R. A., "Studies in Crop Variation. I. An Examination of the Yield of Dressed Grain from Broadbalk," *Journal of Agricultural Science*, vol. 11, pp. 107–135, 1921.

[153] FISHER, R. A., "Studies in Crop Variation. II. The Manurial Response of Different Potato Varieties," *Journal of Agricultural Science*, vol. 13, pp. 311–320, 1923.

[154] FITZGERALD, C. J., WESTON, N. R., PUTNAM, Z.R., AND MAVRIS, D.N., "A Conceptual Design Environment for Technology Selection and Performance Optimization for Torpedoes." Presented at the 9th Multi-Disciplinary Analysis and Optimization Symposium, Atlanta, GA, September 4-6, 2002.

[155] FOWLER, M., *UML Distilled, Third Edition*. Addison-Wesley, 2004.

[156] FRANK, P., "Integrated Extended Air Defence Synthetic Environment (IEADSE)." Presented at the 2005 FLAMES User Group Conference, June 2005.

[157] FRANKLIN, S., AND GRAESSER, A., "Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents," in *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag*, 1996.

[158] Frick, R. K., "Operations Research and Technological Forecasting," *Air University Review*, May-June 1974.

[159] Frits, A. P., *Formulation of an Integrated Robust Design and Tactics Optimization Process for Undersea Weapon Systems.* PhD thesis, Georgia Institute of Technology, 2005.

[160] Fuchs, R. P., et. al., "Why and Whither Hypersonics Research in the U.S. Air Force," tech. rep., United States Air Force Scientific Advisory Board, September 2000.

[161] Garcia, E., *Development of a Framework for the Assessment of Capacity and Throughput Technologies within the National Airspace System.* PhD thesis, Georgia Institute of Technology, January 2003.

[162] Garner, David, "Operation Desert Shield and Desert Storm." Powerpoint Presentation by the Logistics Management Institute, September 1991. The information contained in this presentation has been obtained from only unclassified public access or published sources.

[163] General Accounting Office, "Operation Desert Storm: Evaluation of the Air War." Report to Congressional Requesters, GAO/PEMD-96-10, July 1996.

[164] General Accounting Office, "Operation Desert Storm: Evaluation of the Air Campaign." Report to the Ranking Minority Member, Committee on Commerce, House of Representatives, GAO/NSAID-97-134, June 1997.

[165] General Accounting Office, "Better Management of Technology Development Can Improve Weapon System Outcomes," 1999.

[166] Ghantous, G., "Saudi Arabia Says Will Not Help Any U.S. Strike on Iraq," *Reuters*, 2002.

[167] Gile, R. H., "Global War Game, Second Series 1984-1988." Newport Paper 20, Naval War College, Center for Naval Warfare Studies, 2004.

[168] Gleick, James, *Chaos: Making a New Science.* Penguin Books, 1987.

[169] GlobalSecurity.org, "B-3 Long Range Strike Platform." Online at http://www.globalsecurity.org/military/systems/aircraft/b-3.htm, Last Accessed January 13, 2006.

[170] GlobalSecurity.org, "F/A-22 Raptor." Online at http://www.globalsecurity.org/military/systems/aircraft/f-22.htm, Last Accessed January 13, 2006.

[171] GlobalSecurity.org, "Long Range Strike Aircraft-X Program." Online at http://www.globalsecurity.org/military/systems/aircraft/lrsa.htm, Last Accessed October 5, 2005.

[172] GlobalSecurity.org, "Long Range Strike Platform (LRSP)." Online at http://www.globalsecurity.org/military/systems/aircraft/lrsp.htm, Last Accessed October 5, 2005.

[173] Globalsecurity.org, "Revolutionary Approach To Time Critical Long Range Strike Project (RATTLRS)." Online at http://www.globalsecurity.org/military/systems/munitions/rattlrs.htm. Last Accessed October 26, 2006.

[174] Globalsecurity.org, "Task Force Concepts of Operations (CONOPS)." Online at http://www.globalsecurity.org/military/agency/usaf/tf-conops.htm, Last Accessed April 7, 2006.

[175] GlobalSecurity.org, "X-41 Common Aero Vehicle (CAV)." Online at http://www.globalsecurity.org/space/systems/x-41.htm, Last Accessed January 13, 2006.

[176] GlobalSecurity.org, "B-70 Valkyrie." Online at http://www.globalsecurity.org/wmd/systems/b-70.htm, Last Accessed August 4 2006.

[177] Glosson, B., *War With Iraq: Critical Lessons.* Carolina Gardener, March 2003.

[178] Gonzalez, L., "Spot On: The US Army's There-Based Simulation," *GameSpot.com*, April 21, 2004.

[179] Gordon, M. R., and Trainor, B. E., *The Generals' War.* Little, Brown and Company, 1995.

[180] Gorn, M. H., *Harnessing the Genie: Science and Technology Forecasting for the Air Force 1944-1986.* Office of Air Force History, 1988.

[181] Gorn, M. H., "Technological Forecasting and the Air Force," in *Technology and the Air Force, a Retrospective Assessment*, 1997.

[182] Gotwalt, Christopher, M., "Personal Interview, Cary, NC, June 21, 2006.".

[183] Grant, R., "An Air War Like No Other," *Air Force Magazine*, pp. 30–37, November 2002.

[184] Green J. and Johnson, B., "Towards a Theory of Measures of Effectiveness," *2002 Command and Control Research and Technology Symposium, Naval Postgraduate School, Monterey, California,*, June 11-13, 2002.

[185] Gupta, S. K., Paredis, C. J. J., and Brown, P. F., "Micro Planning for Mechanical Assembly Operations," in *Institute for Complex Engineered Systems, Carnegie Mellon University.*

[186] Haffa, R. P., and Patton, J. H., "Analogues of Stealth." Northrop Grumman Analysis Center Papers, June 2002.

[187] Hale, C. R., "Technololgy Trade Space Development in Crew-Systems for Long-Range Strike." Online at www.hec.afrl.af.mil/Publications/ASC030266.pdf, 2003.

[188] Hall, C. W., *Laws and Models: Science, Engineering and Technology.* CRC Press, 2000.

[189] Hallion, R. P., "Precision Weapons, Power Projection, and the Revolution in Military Affairs." Speech at the United States Air Force Air Armament Summit, Eglin AFB, FL, May 26, 1999.

[190] Hallion, Richard P., *Storm Over Iraq: Airpower and the Gulf War*. Smithsonian Institution Press, 1992.

[191] Hammersley, J. M. and Handscomb, D., *Monte Carlo Methods*. Wiley, New York, 1964.

[192] Hannay, B. and McGinn, R., "The Anatomy of Modern Technology: Prolegomenon to an Improved Public Policy for the Social Management of Technology," *Daedalus*, vol. 109 (Winter), pp. 25–53, 1980.

[193] Hatley, D., Hruschka, P., and Pirbhai, I., *Process for System Architecture and Requirements Engineering*. Dorset House Publishing Company, 2000.

[194] Hause, M., Thom, F., and Moore, A., "The Systems Modelling Language (SysML) (sic)." White Paper by ARTiSAN Software, 2004.

[195] Hawkins, J. and Blakeslee, S., *On Intelligence*. Times Books, 2004.

[196] Hay, B., and Gile, R. H., "Global War Game: The First Five Years." Newport Paper 4, Naval War College, Center for Naval Warfare Studies, June 1993.

[197] Hayes-Roth, B., "An Architecture for Adaptive Intelligent Systems," *Artificial Intelligence: Special Issue on Agents and Interactivity*, vol. 72, pp. 329–365, 1995.

[198] Hazelrigg, G. A., *Systems Engineering: an Approach to Information-Based Design*. Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1996.

[199] Hazlett, J., "T&E M&E War Game Results." Powerpoint Presentation, Online at www.dtic.mil/ndia/2006test/hazlett.pdf. Last Accessed February 12, 2007.

[200] Hebert, A. J., "The Long Reach of the Heavy Bombers," *Air Force Magazine*, November 2003.

[201] Hebert, A. J., "Long Range Strike in a Hurry," *Air Force Magazine*, pp. 26–31, November 2004.

[202] Hebert, A. J., "Strategic Force," *Air Force Magazine*, pp. 38–43, February 2007.

[203] Hebert, A. J., "The Baghdad Strikes," *Air Force Magazine*, vol. 86, no. 7, pp. 46–50, July 2003.

[204] Hebert, A. J., "Compressing the Kill Chain," *Air Force Magazine*, pp. 60–64, March 2003.

[205] Hebert, A. J., "The 2018 Bomber and Its Friends," *Air Force Magazine*, pp. 24–29, October 2006.

[206] Hines, N., *A Probabilistic Methodology for Radar Cross Section Prediction in Conceptual Aircraft Design*. PhD thesis, Georgia Institute of Technology, August 2001.

[207] Holmes, B. J. and Scott, J. M., "Transportation Network Topologies." Submitted for the 4th Integrated Communications, Navigation, and Surveillance (ICNS) Conference, Fairfax, Virginia, April 2004.

[208] Horner, C. A., "Reflections on Desert Storm: The Air Campaign." Powerpoint Presentation.

[209] Hsu, Feng-Hsiung, *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*. Princeton University Press, 2004.

[210] Hwang, C. and Yoon K., *Multiple Attribute Decision Making, Methods and Applications, a State-of-the-Art Survey*. Springer-Verlag, 1981.

[211] IDS Scheer, "ARIS Design Platform Overview." http://www.ids-scheer.com/international/english/products/aris_design_platform/49636.

[212] Ilachinski, A., "Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial-Life Approach to Land Warfare (U)," in *CRM 97-61.1, Center for Naval Analyses*, August 1997.

[213] Ilachinski, A., *Artificial War: Multiagent-Based Simulation of Combat*. World Scientific, 2004.

[214] Ilachinski, A., "Land Warfare and Complexity, Part I: Mathematical Backgorund (sic) and Technical Sourcebook (U)," tech. rep., CIM-461, Center for Naval Analyses, July 1996.

[215] Iman, R. L., Helton, J. C., Campbell, J. E., et. al., "An Approach to Sensitivity Analysis of Computer Models, Part I. Introduction, Input Variable Selection and Preliminary Variable Assessment," *Journal of Quality Technology*, 1981.

[216] Institute of Electrical and Electronics Engineers, "Standard for Application and Management of the Systems Engineering Process." IEEE Standard 1220-1998, 1998.

[217] International Council on Systems Engineering (INCOSE) Technical Board, *Systems Engineering Handbook*. International Council on Systems Engineering, INCOSE-TP-2003-002-03, Version 3.0, June 2006.

[218] Jackson, P., ed., *Jane's All The World's Aircraft*. Jane's Information Group, 2004-2005.

[219] Jaggers, T. J., "Presentation to the House Armed Services Committee, Subcommittee on Terrorism, Unconventional Threats, and Capabilities on Fiscal Year 2007 Air Force Science and Technology." Report to the United States House of Representatives, online at www.house.gov/hasc/3-29-06JaggersTestimony.pdf, March 29, 2006.

[220] JCS J-8/Force Application Assessment Division, "White Paper on Conducting a Capabilities-Based Assessment (CBA) Under the Joint Capabilities Integration and Development System (JCIDS)." Online at www.dtic.mil/futurejointwarfare/strategic/cba_guide06.pdf, January 2006.

[221] Jeffrey, R., *Probabilistic Thinking*. Princeton University Online Text, 1999.

[222] Jennings, N., "On Agent-Based Software Engineering," *Artificial Intelligence*, no. 117, pp. 227–296, 2000.

[223] Jimenez, H., Mavris, D.N., "Conceptual Design of Current Technology and Advanced Concepts for an Efficient Multi-Mach Aircraft." Presented at the World Aviation Congress, Dallas, TX, 2005.

[224] Johnson, C., "Introduction to Neural Networks in JMP, Version 2.0," tech. rep., Georgia Institute of Technology, 2004.

[225] Johnson, C., and Schutte, J., *Basic Regression Analysis for Integrated Neural Networks (BRAINN) Documentation, Version 1.2.* Georgia Institute of Technology, 225 North Avenue, Atlanta, GA 30332, June 14 2005.

[226] Johnson, F., "Air Force Research Lab Multi-Directorate Vision: Sensor Craft." Information Sheet, Approved for Public Release, 2005.

[227] Johnson, M., Moore, L., and Ylvisaker, D., "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, vol. 26, pp. 131–148, 1990.

[228] Jones, L. G. and Lattanze, A. J., "Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study." Technical Note CMU/SEI-2001-TN-022, December 2001. Published Under the Architecture Tradeoff Analysis Initiative.

[229] Jones, W. F. and Schmisseur, J. D., "Aeronautics at AFRL." Powerpoint Presentation to the Aeronautics and Space Engineering Board, National Research Council, November 9 2005.

[230] Jumper, J. P., "Global Strike Task Force," *Aerospace Power Journal*, Spring 2001.

[231] Jumper, J. P. and Roche, J. G., "Air Force 2002 Posture Statement," tech. rep., US Department of Defense, 2002.

[232] Jumper, J. P., "Speech at the Air Force Association 17th Annual Air Warfare Symposium." Online at http://www.afa.org/AEF/pub/Jumper2001.asp, Orlando, FL., February 15, 2001.

[233] Kass, R. A., *The Logic of Warfighting Experiments.* United States Department of Defense Command and Control Research Program (CCRP), 2006.

[234] Kaufmann, A., *The Science of Decision Making, an Introduction to Praxeology.* World University Library, 1968.

[235] Keating, C., et. al., "System of Systems Engineering," *Engineering Management Journal*, vol. 15, no. 3, pp. 36–45, September 2003.

[236] Khalilzad, Z., Ochmanek, D., and Shapiro, J., "Forces for What? Geopolitical Context and Air Force Capabilities," in *United States Air and Space Power in the 21st Century*, 2002.

[237] Kim, H. M., and Hidalgo, I. J., "System of Systems Optimization by Pseudo-Hierarchical Multistage Model." AIAA 2006-6921, Presented at the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA., September 6-8, 2006.

[238] King, J., Ensor, D., et. al., "'Decapitation Strike' Was Aimed at Saddam." Online at http://www.cnn.com/2003/WORLD/meast/03/20/sprj.irq.target.saddam/, March 20, 2003.

[239] Kirby, M., "A Methodology for Technology Identification Evaluation and Selection for Complex Systems." Powerpoint Presentation, 2003.

[240] Kirby, M. R., *A Methodology for Technology Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design*. PhD thesis, Georgia Institute of Technology, March 2001.

[241] Kirby, M. R., Rupp, J., and Mavris, D. N., "An Approach for the Strategic Planning of Future Technology Portfolios." SAE 2004-01-3109, Presented at the World Aviation Congress, Reno, NV, November 2-4, 2004.

[242] Klann, J. L., Snyder, C. A., "NASA Engine Performance Program Users Manual." NASA Lewis Research Center, Cleveland, Ohio, March 1997.

[243] Kline, M., *Mathematics: The Loss of Certainty*. Oxford University Press, 1980.

[244] Koch, P. N., Mavris, D. N., and Mistree, F., "Multi-Level, Partitioned Response Surfaces for Modeling Complex Systems." Presented at the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, September 2-4, 1998.

[245] Koeppel, D., "Massive Attack," *Popular Science*, July 2003.

[246] Kopp, C., "Air Power Australia." Online at http://www.ausairpower.net, Last Accessed November 3, 2005.

[247] Kopp, C., "Lockheed-Martin F-35 Joint Strike Fighter." Online at http://www.ausairpower.net/jsf-analysis-2002.html, Last Accessed December 13, 2005.

[248] Korn, S., "Munitions Technology Update." Presented at the National Defense Industry Association 30th Air Armament Symposium, Online at www.ndiagulfcoast.com/events/archive/30th_symposium/day2/Korn/ndia-mr-korn.pdf, October 12-13, 2004.

[249] Krepinevich, A., Watts, B., and Work, R., "Meeting the Anti-Access and Area-Denial Challenge," tech. rep., Center for Strategic and Budgetary Assessments, 2003.

[250] Krygiel, Annette J., *Behind the Wizard's Curtain: An Integration Environment for a System of Systems*. DoD C4ISR Cooperative Research Program, 1999.

[251] Kumpel, A. E., Barros, P. A., and Mavris, D. N., "A Quality Engineering Approach to the Determination of the Space Launch Capability of the Peacekeeper ICBM Utilizing Probabilistic Methods." Presented at the Missile Sciences Conference, Monterey, CA, November 5-7, 2002.

[252] Kurzke, Joachim, "GasTurb 10: A Program for Gas Turbine Performance Calculations." Computer Program, Online at http://www.gasturb.de, 2004.

[253] Lambeth, B. S., *The Transformation of American Air Power.* Cornell University Press, 2000.

[254] Lambeth, B. S., *NATO's Air War for Kosovo: A Strategic and Operational Assessment.* RAND Corporation, MR-1365-AF, 2001.

[255] Lanicci, J. M., "Weather Operations in the Transformation Era." Air War College, Maxwell Paper No. 29, March 2003.

[256] Leahy, M. B., "Control Challenges for the Next Century of Flight." Plenary Address at the 2005 American Control Conference, Portland, OR, June 2005.

[257] Leemis, L. M., *Reliability: Probabilistic Models and Statistical Methods.* Prentice Hall, 1995.

[258] Lewe, J., *An Integrated Decision-Making Framework for Transportation Architectures: Application to Aviation Systems Design.* PhD thesis, Georgia Institute of Technology, 2005.

[259] Lewe, J., Upton, E., Mavris, D., and Schrage, D., "An Agent-Based Framework for Evaluating Future Transportation Architectures." AIAA-2003-6769, Presented at the AIAA 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum, Denver, Colorado, Nov. 17-19, 2003.

[260] Lewin, R., *Complexity, Life at the Edge of Chaos.* Collier Books, 1992.

[261] Lockheed Martin Missiles and Fire Control, "LOCAAS: Low Cost Autonomous Attack System." Information Sheet, 2002.

[262] Lorenz, E. Paper to the New York Academy of Sciences, 1963.

[263] Luce, B. R., *Medical Technology and its Assessment.* Del Mar Publishers, Albany, New York, 1993.

[264] Madyastha, V., *Adaptive Estimation for Control of Uncertain Nonlinear Systems with Applications to Target Tracking.* PhD thesis, Georgia Institute of Technology, 2006.

[265] Maes, P., "Artificial Life Meets Entertainment: Life like Autonomous Agents," *Communications of the ACM*, vol. 38, no. 11, pp. 108–114, 1995.

[266] Mahafza, S., Componation, P., and Tippett, D., "A Performance-Based Technology Assessment Methodology to Support DoD Acquisition," *Defense Acquisition Review Journal*, pp. 269–283.

[267] Maier, M. W., "Architecting Principles for Systems of Systems," in *Proceedings of the Sixth Annual International Symposium, International Council on Systems Engineering*, Boston, MA, 1996.

[268] Maier, M. W., and Eberhardt, R., *The Art of Systems Architecting, Second Edition.* CRC Press, 2000.

[269] MANUCY, A., ed., *Artillery Through the Ages.* National Park Service Interpretive Series History No. Three, Washington, D.C., 1949.

[270] MARSAW, A., GERMAN, B., HOLLINGSWORTH, P., MAVRIS, D., AND CSONKA, S., "An Interactive Visualization Environment for Decision Making in Aircraft Engine Preliminary Design." AIAA-2007-1333, Presented at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 8-11, 2007.

[271] MARSDEN, E. W., *Greek and Roman Artillery.* Oxford University Press, 1969.

[272] MARZOLF, G. S., "Time-Critical Targeting, Predictive Versus Reactionary Methods: An Analysis for the Future." Air University Press, College of Aerospace Doctrine, Research and Education (CADRE) Paper No. 19, September 2004.

[273] MATCHWARE, "OpenMind Brainstorming Software." http://www.matchware.net/en/products/openmind/default.htm, Last Accessed August 3, 2005.

[274] MATHWORKS, "Model-Based Calibration Toolbox Version 3." Computer Program, MathWorks, Inc., November 2005.

[275] MATHWORKS, "Creating Space-Filling Designs." Online at http://www.mathworks.com/access/helpdesk/help/toolbox/mbc/mbc_gs/f3-7640.html, Last Accessed July 22, 2006.

[276] MATSUMURA, J., STEEB, R., GORDON, J., AND STEINBERG, P., "Preparing for Future Warfare with Advanced Technologies." Rand Corporation, Issue Paper 215, 2002.

[277] MATTINGLY, J. D., "Data for Some Military Turbofan Engines." Online at http://www.aircraftenginedesign.com/TableB2.html, Last Accessed May 15, 2006.

[278] MATTINGLY, J. D., HEISER, W. H., AND PRATT D. T., *Aircraft Engine Design, Second Edition.* AIAA Education Series, 2002.

[279] MAVRIS, D., "Application of Parametric Analysis to Aircraft Bus Timing ," tech. rep., Georgia Institute of Technology, School of Aerospace Engineering, 2004.

[280] MAVRIS, D., "Architecture-Based Modeling and Simulation in Support of the Accelerated Capabilities Development and Delivery (ACD&D) Program." Final Report, December 31, 2004.

[281] MAVRIS, D., "Multi-Disciplinary Design Optimization Support for Surface Ship Projects," tech. rep., Georgia Institute of Technology, School of Aerospace Engineering, 2004.

[282] MAVRIS, D., "Multidisciplinary Optimization of Naval Ship Design and Mission," tech. rep., Georgia Institute of Technology, School of Aerospace Engineering, 2004.

[283] MAVRIS, D. N., "Work Plan for Ultra Efficient Engine Technology Systems Evaluation," tech. rep., School of Aerospace Engineering, Georgia Institute of Technology, 2003.

[284] Mavris, D. N., "FY03 UEET Metrics Assessment and Parametric Exploration," tech. rep., Georgia Institute of Technology, School of Aerospace Engineering, 2004.

[285] Mavris, D. N. and Kirby, M. R., "Technology Identification, Evaluation, and Selection for Commercial Transport Aircraft." Presented at the 58th Annual Conference Of Society of Allied Weight Engineers, May 1999.

[286] Mavris, D. N., Baker, A. P., Schrage, D. P., "Technology Infusion and Resource Allocation for a Civil Tiltrotor." Proceedings of the AHS Vertical Lift Aircraft Design Conference, San Francisco, CA, January 19-21, 2000.

[287] Mavris, D. N., Baker, A. P., Schrage, D. P., "Simultaneous Assessment of Requirements and Technologies in Rotorcraft Design." Presented at the 56th Annual Forum of the American Helicopter Society, Virginia Beach, VA, May 2-4, 2000.

[288] Mavris, D. N., Bandte, O., "Comparison of Two Probabilistic Techniques for the Assessment of Economic Uncertainty." Presented at the 19th Annual Conference of the International Society of Parametric Analysts, New Orleans, LA, May 1997.

[289] Mavris, D. N., Biltgen P. T., Ender, T. R., and Cole, B., "Technology Assessment and Capability Tradeoff Using Architecture-Based Systems Engineering Methodologies." Presented at the 1st International Conference on Innovation and Integration in Aerospace Sciences, Queens University Belfast, Northern Ireland, UK., August 4-5, 2005.

[290] Mavris, D. N, Briceno, S. I., Buonanno, M., Fernandez, I., "A Parametric Exploration of Supersonic Business Jet Concepts Utilizing Response Surfaces." Presented at the 2nd AIAA ATIO Forum, Los Angeles, CA, October 1-3, 2002.

[291] Mavris, D. N., Collins, K. B., Schrage, D. P., "A Method of Qualitative Analysis During Conceptual Design as Applied to Unmanned Aerial Vehicles(A)." Presented at the American Helicopter Society 60th Annual Forum, Baltimore, MD, June 7-10, 2004.

[292] Mavris, D. N., DeLaurentis, D. A., Bandte, O., Hale, M. A., "A Stochastic Approach to Multi-disciplinary Aircraft Analysis and Design." AIAA-98-0912, Presented at the 36th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 12-15, 1998.

[293] Mavris, D. N., Mantis, G., Kirby, M. R., "Demonstration of a Probabilistic Technique for the Determination of Economic Viability." SAE-975585, Presented at the 2nd World Aviation Congress and Exposition, Anaheim, CA, October 13-16, 1997.

[294] Mavris, D. N., Roth, B. A., Macsotai, N. I., "A Method for Probabilistic Sensitivity Analysis of Commercial Aircraft Engines." Presented at the 14th ISABE, Florence, Italy, September 1999.

[295] Mavris, D. N., Soban, D. S., Largent, M. C., "An Application of a Technology Impact Forecasting (TIF) Method to an Uninhabited Combat Aerial Vehicle." Presented at the 4th World Aviation Congress and Exposition, San Francisco, CA, October 19-21, 1999.

[296] McClure, E. K., *An Evolving-Requirements Technology Assessment Process for Advanced Propulsion Concepts.* PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, 2006.

[297] McCulloch, W. S. and Pitts, W. H., "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[298] McDonald, R., and Mavris, D., "Formulation, Realization, and Demonstration of a Process to Generate Aerodynamic Metamodels for Hypersonic Cruise Vehicle Design." Presented at the 4th World Aviation Congress and Exposition, San Francisco, CA, October 10-12, 2000, SAE/AIAA 2000-01-5559.

[299] MGM Studios, "Wargames." Motion Picture, 1983.

[300] Montgomery, D., *Design and Analysis of Experiments, 5th Edition.* Wiley & Sons, New York, 2001.

[301] Morehouse, J., "Time Critical Targeting." Powerpoint Presentation by the Headquarters, U.S. Air Force.

[302] Morokoff, W.J., and Caflisch, R.E., "Quasi-Random Sequences and Their Discrepancies," *SIAM Journal of Scientific Computing*, vol. 15, no. 6, pp. 1251–1279, 1994.

[303] Morse, P. M. and Kimball, G. E., *Methods of Operations Research.* OEG Report 54, Office of the Chief of Naval Operations, Navy Department, Washington, D.C., 1946.

[304] Moseley, M. T., "Operation IRAQI FREEDOM - By The Numbers," tech. rep., CENTAF Assessment and Analysis Division, April 30, 2003.

[305] Moseley, M. T. and Wynne, M. W., "United States Air Force Posture Statement," March 2006.

[306] Myers, R. B., "The National Military Strategy of the United States of America, A Strategy for Today; A Vision for Tomorrow," 2004.

[307] National Academy of Sciences, *Modeling Human and Organizational Behavior, Application to Military Simulations.* Published by the National Academies Press, Panel on Modeling Human Behavior and Command Decision Making: Representations for Military Simulations, National Research Council, 1998.

[308] National Academy of Sciences, *Modeling and Simulation in Manufacturing and Defense Acquisition: Pathways to Success.* Published by the National Academies Press, Committee on Modeling and Simulation Enhancements for 21st Century Manufacturing and Defense Acquisition, National Research Council, 2002.

[309] National Academy of Sciences, *Effects of Nuclear Earth-Penetrator and Other Weapons.* Published by the National Academies Press, Committee on the Effects of Nuclear Earth-Penetrator and Other Weapons, National Research Council, 2005.

[310] National Academy of Sciences, *Defense Modeling, Simulation, and Analysis: Meeting the Challenge*. Published by the National Academies Press, Committee on Modeling and Simulation for Defense Transformation, National Research Council, 2006.

[311] National Academy of Sciences, *Future Air Force Needs for Survivability*. Published by the National Academies Press, Committee on Future Air Force Needs for Survivability, National Research Council, 2006.

[312] National Aeronautics and Space Administration, "NASA Systems Engineering Handbook." SP-610S, June 1995.

[313] National Aeronautics and Space Administration, "Space Shuttle STS-99 Press Kit." Online at http://www.shuttlepresskit.com/sts-99/OBJ107.htm, January 18, 2000. Last Accessed February 13, 2006.

[314] National Aeronautics and Space Administration, "Numerical Propulsion System Simulation (NPSS) User Gude, Version 1.6.3." Online at http://virtualtestcell.com/docs/UserGuide.doc, April 4, 2005.

[315] National Geospatial Intelligence Agency, "NGA Map Images." Online at www.globalsecurity.org, Last Accessed April 2006.

[316] National Geospatial Intelligence Agency, "NGA Raster Roam, DTED Level 0." Online at http://geoengine.nima.mil/geospatial/SW_TOOLS/NIMAMUSE/webinter/rast_roam.html, Last Accessed March 3, 2006.

[317] New Line Cinema, "The Lord of the Rings - The Two Towers (Platinum Series Special Extended Edition)." Motion Picture, December 2002.

[318] NIST/SEMATECH, "e-Handbook of Statistical Methods." Online at http://www.itl.nist.gov/div898/handbook/, Updated July 18, 2006. Last Accessed August 2006.

[319] Nitze, P. H. and McCall, J. H., "Contemporary Strategic Deterrence and Precision-Guided Munitions," in *Post-Cold War Conflict Deterrence*, National Academy Press, Washington, D.C., 1997.

[320] Nixon, J. and Mavris, D., "A Multi-Level, Hierarchical Approach to Technology Selection and Optimization." Presented at the 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA, Atlanta, GA, September 4-6, 2002.

[321] Noble, J. S. and Tanchoco, J. M. A., *Concurrent Engineering: Automation, Tools, and Techniques*, ch. Design for Economics, pp. 401–436. John Wiley and Sons, 1993.

[322] Obering, H., "Missile Defense Program and Fiscal Year 2006 Budget." Online at http://www.mda.mil/mdalink/html/statements.html, Spring 2005.

[323] Oberkampf, W. L., DeLand, S. M., Rutherford, B. M., Diegert, K. V., Alvin, K. F., "A New Methodology for the Estimation of Total Uncertainty in Computational Simulation." AIAA-99-1612, 1999.

[324] Oeltjen, C. L., "A Comparison of Computational Cognitive Models: Agent-Based Systems Versus Rule-Based Architectures," Master's thesis, Naval Postgraduate School, 2003.

[325] Office of the Assistant Secretary of the Navy for Research Development and Acquisition, "Naval Capability Evolution Process Guidebook, Volume I," May 23, 2005.

[326] Office of the Under Secretary of Defense For Acquisition, Technology, and Logistics, "Report of the Defense Science Board Task Force on Future Strategic Strike Forces," tech. rep., Defense Science Board, February 2004.

[327] O'Halloran, J. and Foss, C., eds., *Jane's Land-Based Air Defence.* Jane's Information Group, 2004-2005.

[328] Oliver, N. and Intille, S., "Modeling Audience Group Behavior." MIT Media Lab, http://web.media.mit.edu/̃intille/audience/audience.html, October 2004. Last Accessed May 2006.

[329] Olson, E. D. and Mavris, D. N., "Development of Response Surface Equations for High-Speed Civil Transport Takeoff and Landing Noise." Presented at the 2nd World Aviation Congress and Exposition, Anaheim, CA, October 13-16, 1997.

[330] O'Malley, William D., *Evaluating Possible Airfield Deployment Options: Middle East Contingencies.* RAND Corporation, MR-1353-AF, 2001.

[331] Onley, D. S., "Air Force Unveils New IT-Centric Strategy," *Government Computer News*, July 29, 2002.

[332] Owen, G., *Game Theory, 3rd Edition.* Academic Process, 2001.

[333] Painter, R. D., "Object-Oriented Military Simulation Development and Application," in *Proceedings of the 1995 Winter Simulation Conference* (Alexopoulos, C., Kang, K., Lilegdon, W. R., and Goldsman, D., ed.), 1995.

[334] Parsch, A., "Nothrop Grumman GAM (GPS-Aided Munition)." Online at http://www.designation-systems.net/dusrm/app5/gam.html, Last Accessed March 3, 2006.

[335] Partin, J. W., ed., *A Brief History of Fort Leavenworth.* Combat Studies Institute, 1983.

[336] Peters, R., "Air Force Integrated Collaborative Environment (AF-ICE) Program Update." Powerpoint Briefing, August 2005.

[337] Picton, H., *Introduction to Neural Networks.* Macmillan Press, 1994.

[338] Pike, J., "Global Security Homepage." Online at www.globalsecurity.org.

[339] Plenge, B. T., "Air Force Research Laboratory Area Dominance Munition Technology." Information Sheet, AAC/PA 10-13-05-406. Obtained January 2007.

[340] Plenge, B. T., "Area Dominance," *AFRL Technology Horizons*, vol. 5, no. 2, April 2004.

[341] Plenge, B. T., "Military Worth Analysis of New Concept Weapons," *AFRL Technology Horizons*, vol. 5, no. 6, April 2006.

[342] Pohlmann, L. D., "Is Systems Engineering for Systems-of-Systems Really Any Different?." Powerpoint Presentation at the INCOSE International Symposium, Orlando, Florida, July 11, 2006.

[343] Polya, H., *How to Solve It, 2nd Edition.* Doubleday-Anchor Books, Garden City, N.J., 1957.

[344] Pounds, L. and Allen, K., "Beyond Mission Level." Applied Research Associates, Presented at the 2005 FLAMES User's Conference, June 2005.

[345] Powell, S. M., "Scud War, Round Three," *Air Force Magazine*, vol. 75, no. 10, October 1992.

[346] Powell, W., "General Jumper Qualifies to Fly Air Force's Newest Fighter." Air Force Education and Training Command News Service, Online at http://www.aetc.randolph.af.mil/pa/AETCNS/Jan2005/011305009.htm, January 13, 2005. Last Accessed August 5, 2006.

[347] Princeton University Cognitive Science Laboratory, "WordNet, A Lexical Database for the English Language." Online at http://wordnet.princeton.edu/, Last Accessed October 2005.

[348] Putney, D. T., *Airpower Advantage: Planning the Gulf War Air Campaign 1989-1991.* Air Force History and Museums Program, United States Air Force, 1994.

[349] Rao, M., "DoDAF with UML/SysML and ARTiSAN Studio." White Paper by ARTiSAN Software, 2005.

[350] Ritchey, T., "General Morphological Analysis, A General Method for Non-Quantified Modelling." Adapted from an paper presented at the 16th EURO Conference on Operational Analysis, Brussels, 1998.

[351] Ritter, N. and Ruth, M., "GeoTIFF Format Specification, Revision 1.0." Online at http://www.remotesensing.org/geotiff/spec/geotiffhome.html, December 28, 2000.

[352] Robbins, J. S., "The Ultimate Shock and Awe." National Review, online at http://www.nationalreview.com/robbins/robbins032103.asp, March 21, 2003. Last Accessed May 5, 2006.

[353] Roche, J. G., and Watts, B. D., "Choosing Analytic Measures," *Journal of Strategic Studies*, vol. 14, pp. 165–209, 1991.

[354] Rodrigues, L. J., "B-2 Bomber Cost and Operational Issues." GAO/NSIAD-97-181, United States General Accounting Office Report to Congressional Committees, August 1997.

[355] Roseneau, W., "Special Operations Forces and Elusive Enemy Ground Targets: Lessons from Vietnam and the Persian Gulf War." RAND Corporation Monograph, MR-1408-AF, 2001. ISBN: 0-8330-3071-X.

[356] Roskam, J., *Airplane Design Part I: Preliminary Sizing of Airplanes*. Design Analysis and Research (DAR) Corporation, June 1989.

[357] Rostker, B., "Information Paper: Iraq's Scud Ballistic Missiles." Online at http://www.gulflink.osd.mil/scud_info, July 25, 2000. Published by the Special Assistant for Gulf War Illness, Department of Defense. Last Accessed August 13, 2006.

[358] Roth, B., Mavris, D., and Elliott, D., "A Probabilistic Approach to UCAV Engine Sizing." Presented at the 34th Joint Propulsion Conference, Cleveland, OH, July 13-15, 1998.

[359] Rumsfeld, D., "Requirements System." Memo to the Chairman of the Joint Chiefs of Staff (General Pace), March 18, 2002.

[360] Rumsfeld, D., "The National Defense Strategy of the United States of America," March 2005.

[361] Rumsfeld, D. and Myers, R., "DoD News Briefing - Secretary Rumsfeld and Gen. Myers." Online at http://www.yale.edu/lawweb/avalon/sept_11/dod_brief25.htm, October 8, 2001. Last Accessed January 27, 2007.

[362] Russell, J., *The Big Issue: Command and Combat for the Information Age*. Strategic and Combat Studies Institute, The Occasional, Number 45, 2003.

[363] Russell, S. J. and Norvig, P., *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, N.J., Prentice Hall, 1995.

[364] Ruth, M., "GeoTIFF FAQ Version 2.3." Online at http://www.remotesensing.org/geotiff/faq.html, Last Accessed February 2005.

[365] Ryan, R. S., and Townsend, J. S., "Application of probabilistic analysis and design methods in space programs," *Journal of Spacecraft and Rockets*, vol. 31, no. 6, pp. 1038–1043, 1994.

[366] Sall, J., et al., *JMP Design of Experiments*. SAS Institute, Inc., Cary, N.C., 2005.

[367] Scheffe, H., *The Analysis of Variance*. John Wiley and Sons, 1959.

[368] Schrage, D., "Technology for Rotorcraft Affordability Through Integrated Product/Process Development (IPPD)." Alexander A. Nikolsky Lecture, Presented at the American Helicopter Society 55th Annual Forum, Montreal, Canada, May 25th-29th, 1999.

[369] Schrage, D. P., *Concurrent Engineering: Automation, Tools, and Techniques*, ch. Concurrent Design: A Case Study, pp. 535–582. John Wiley and Sons, 1993.

[370] Schwartz, N. A., "Joint Capabilities Integration and Development System." Chairman of the Joint Chiefs of Staff Instruction CJCSI 3170.01E, May 11, 2005.

[371] Scott, W. B., "Vision Takes Form," *Aviation Week and Space Technology*, p. 22, August 15, 2005.

[372] Shelton, H. H., "Joint Doctrine Capstone and Keystone Primer," tech. rep., United States Department of Defense Joint Chiefs of Staff, 2001.

[373] Shlapak, D. A., Stillion, J., Oliker, O., and Charlick-Paley, T., *A Global Access Strategy for the U.S. Air Force*. RAND Corporation, Project Air Force, MR-1216-AF, 2002.

[374] Simmons, E., "A Capabilities-Based Approach to Force Planning." Powerpoint Presentation, Online at www.ccc.nps.navy.mil/events/recent/simmonsOct04ppt.pdf, September 21, 2004.

[375] Simon, H. A., "Artificial Intelligence: Where Has it Been and Where is it Going?," *IEEE Transactions on Knowledge and Data Engineering*, vol. 3, pp. 192–136, June 1991.

[376] Smyth, J., "MC2A: Multi-Sensor Command and Control Aircraft." Powerpoint Presentation, Online at http://esc.hanscom.af.mil/esc-mc2a/, Last Accessed January 20, 2006.

[377] Soban, D. S. and Mavris, D. N., "The Need for a Military System Effectiveness Framework - The System of Systems Approach." AIAA-2001-5226, Presented at the 1st Aircraft, Technology Integration, and Operations Forum, Los Angeles, CA., Oct. 16-18, 2001.

[378] Soban, D. S., *A Methodology for the Probabilistics Assessment of System Effectiveness as Applied to Aircraft Survivability and Susceptibility*. PhD thesis, Georgia Institute of Technology, 2001.

[379] Southwest Research Institute, "FPI User's and Theoretical Manual," San Antonio, TX, 1995.

[380] Spick, M., *The Great Book of Modern Warplanes*. Salamander Books Limited, London, 2000.

[381] Sproles, N., "Formulating Measures of Effectiveness," *Systems Engineering*, vol. 5, no. 4, pp. 253–263, 2002.

[382] Sproles, N., "Measures of Effectiveness; How Will I Recognise That I Have Succeeded?," in *Proceedings of the INCOSE UK Fourth Annual Symposium, Hendon, UK*, pp. 95–102, June 1-2, 1998.

[383] Stanley, J., *Introduction to Neural Networks*. Scientific Software, Pasadena, CA, 1990.

[384] Starr, B., "'Bunker Busters' May Grow to 30,000 Pounds." Published by CNN.com, online at http://www.cnn.com/2004/US/07/20/big.bomb/index.html, July 21, 2004. Last Accessed April 15, 2006.

[385] Stevens, D., Gibson, J., and Ochmanek, D., "Modernizing the Combat Forces: Near-Term Options," in *United States Air and Space Power in the 21st Century*, 2002.

[386] Stonier, R. A., "Stealth Aircraft and Technology From World War II to the Gulf," *SAMPE Journal*, vol. 27, no. 4, pp. 9–17, 1991.

[387] SysML Object Management Group, "UML for Systems Engineering Request for Proposal, OMG Document: ad/03-03-41." Online at www.sysml.org, September 2003.

[388] SysML Partners (www.sysml.org), "Systems Modeling Language (SysML) Specification, Version 0.9," tech. rep., January 10, 2005.

[389] Systems and Software Consortium, "Architecture." Online at http://www.software.org/pub/architecture/dodaf.asp, Last Accessed October 2005.

[390] Systems and Software Consortium, "Zachman Framework." Online at http://www.software.org/pub/architecture/zachman-frame.asp, Last Accessed March 2006.

[391] Taguchi, G., *Introduction to Quality Engineering: Designing Quality into Products and Processes*. Hong Kong: Asian Productivity Organization, 1986.

[392] Tai, J. C., Mavris, D. N., and Schrage, D. P., "An Application of Response Surface Methodology to the Design of Tipjet Driven Stopped Rotor/Wing Concepts." Presented at the 1st AIAA Aircraft Engineering, Technology, and Operations Congress, Anaheim, CA., September 19-21, 1995.

[393] Takagi, H., "Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation," in *Proceedings of the IEEE*, vol. 89, pp. 1275–1296, 2001.

[394] Tangen, S., "Development of an Integrated Environment for Aircraft Sizing, Mission Planning, and Capability Analysis." AE8900 Special Topics Report, Georgia Institute of Technology, School of Aerospace Engineering, December 2006.

[395] Tejtel, D., Zeune, C. H., et. al., "Breathing New Life into Old Processes: An Updated Approach to Vehicle Analysis and Technology Assessment." Presented at the AIAA 5th Aviation, Technology, Integration, and Operations Conference (ATIO), September 26-28, 2005.

[396] Temple, L. P. III, *Shades of Gray: National Security and the Evolution of Space Reconnaissance*. American Institute of Aeronautics and Astronautics, Reston, VA, 2005.

[397] Ternion Corporation, "FLAMES Overview." Online at http://www.ternion.com/. Last Accessed October 17, 2005.

[398] Ternion Corporation, "FLAMES Customization Guide, Version 6.0," 2005.

[399] Ternion Corporation, "FLAMES Sensor Coverage Option Manuall, Version 6.0," 2005.

[400] Tesfatsion, L., "Agent-Based Computational Economics, Growing Economies from the Bottom Up." Department of Economics, Iowa State University, http://www.econ.iastate.edu/tesfatsi/ace.htm, October 2004. Last Accessed November 2005.

[401] The Boeing Corporation, "B-52 Stratofortress Background Info." http://www.boeing.com/defense-space/military/b52-strat/b52info.html. Last Accessed October 27, 2006.

[402] The Boeing Corporation, "U.S. Air Force B-2 Bomber Drops 80 JDAMs in Historic Test." Press Release, Online at http://www.boeing.com/news/releases/2003/q3/nr_030917o.html, September 17, 2003. Accessed October 27, 2006.

[403] Thompson, L., "Searching for the Next B-52," *Armed Forces Journal*, September 2006.

[404] Thompson, W. E., "F-117 Goes to War," *Flight Journal*, June 2001.

[405] Thoreau, H. D., *Walden; or, Life in the Woods*. Reprint, Dover Publications, 1854, Reprint 1995.

[406] Thurston, D. L., and Carnahan, J. V., *Concurrent Engineering: Automation, Tools, and Techniques*, ch. Intelligent Evaluation of Designs for Manufacturing Cost, pp. 437–462. John Wiley and Sons, 1993.

[407] Tirpak, J. A., "Find, Fix, Track, Target, Engage, Assess," *Air Force Magazine*, July 2000.

[408] Tirpak, J. A., "Bomber Questions," *Air Force Magazine*, pp. 37–43, September 2001.

[409] Tirpak, J. A., "Long Arm of the Air Force," *Air Force Magazine*, pp. 28–34, October 2002.

[410] Tirpak, J. A., "Raptor as Bomber," *Air Force Magazine*, pp. 28–33, January 2005.

[411] Tirpak, J. A., "The Four-Year Sprint," *Air Force Magazine*, pp. 40–43, October 2005.

[412] Titus, N., "Air Force CONOPS and Capabilites Based Planning." Powerpoint Presentation, March 19, 2004.

[413] Towers, J., "Directed Energy and FLAMES." Aegis Technologies, Presented at the 2005 FLAMES User's Conference, June 2005.

[414] Tumer, K. and Wolpert, D., eds., *Collectives and the Design of Complex Systems*. Springer, 2004.

[415] Twiss, B. C., *Forecasting for Technologists and Engineers*. Peter Peregrinus, Ltd., London, U.K., 1992.

[416] Tzu, S., *The Art of War*. 500 B.C.

[417] Ubisoft, Inc., "Lock-On: Modern Air Combat." Online at http://lomac.com/screens.php?id=773, Last Accessed January 2006.

[418] Ueberhuber, C. W., *Numerical Computation 2: Methods, Software, and Analysis*. Berlin: Springer-Verlag, pp. 124-125, 1997.

[419] Under Secretary of Defense for Acquisition Technology, "DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A," May 2003.

[420] United Nations Security Council, "Resolution 678 (1990)." Online at http://www.fas.org/news/un/iraq/sres/sres0678.htm, November 29, 1990. Last Accessed July 17, 2006.

[421] United Nations Special Commission (UNSCOM), "Major Sites Associated With Iraq's Past WMD Programs." Online at http://www.globalsecurity.org/wmd/library/news/iraq/un/971203_sites.htm, December 3, 1997. Last Accessed July 25, 2006.

[422] United States Air Force , "Fact Sheet: B-52 Stratofortress." http://www.af.mil/factsheets/factsheet.asp?fsID=83, Last Accessed August 2005.

[423] United States Air Force, "Air force link: Factsheets." Online at http://www.af.mil/library/factsheets/. Last Accessed October 2006.

[424] United States Air Force, "Air Force Link: History Milestones (1990's)." Online at http://www.af.mil/history/. Last Accessed June 2006.

[425] United States Air Force, "Modeling and Simulation Resource Repository." Online at http://afmsrr.afams.af.mil/. Accessed October 15, 2005.

[426] United States Air Force, "A Ballistic Missile Detection Support System." General Operational Requirement No. 96, June 10, 1955.

[427] United States Air Force, "Targeting: The Joint Targeting Processs and Procedures for Targeting Time-Critical Targets." Air Force Joint Pamphlet (AFJPAM) 10-225, July 1, 1997.

[428] United States Air Force, "Air Force Task List (AFTL)." Air Force Doctrine Document 1-1, August 12, 1998.

[429] United States Air Force, "USAF Intelligence Targeting Guide." Air Force Pamphlet 14-210, February 1, 1998.

[430] United States Air Force, "Organization and Employment of Aerospace Power: Air Force Doctrine Document 2." Online at http://www.e-publishing.af.mil, February 17, 2000.

[431] United States Air Force, "Air Force Basic Doctrine: Air Force Doctrine Document 1." Online at http://www.e-publishing.af.mil, November 17, 2003. Last Accessed October 2006.

[432] United States Air Force, "Wargaming." Air Force Instruction 10-2305, August 15, 2003.

[433] United States Air Force, "Leadership and Force Development: Air Force Doctrine Document 1-1." Online at http://www.e-publishing.af.mil, February 18, 2004. Last Accessed June 2006.

[434] United States Air Force, "Transformation Flight Plan," tech. rep., HQ USAF/XPXC Future Concepts and Transformation Division, 2004.

[435] United States Air Force, "Air Force Instruction 36-110, Air Force Scientific Advisory Board," Last Accessed March 18, 2005.

[436] United States Air Force, "F-16 Operating Procecures: Flying Operations." Air Force Instruction 11-2F-16, Volume 3, Last Accessed September 30, 2005.

[437] United States Air Force, "2007 Budget Request, Next Generation Long Range Strike." Online at www.dtic.mil/descriptivesum/Y2007/AirForce/0604015F.pdf, Last Accessed February 2006.

[438] United States Air Force, "Air Force Link." Online at http://www.af.mil/main/welcome.asp, Last Accessed January 23, 2006.

[439] United States Air Force, "Airman: The Book." Online at http://www.af.mil/news/airman/0106/, Winter 2006. Accessed October 28, 2006.

[440] United States Air Force, "USAF Fact Sheet: E-3 Sentry (AWACS)." Online at http://www.af.mil/factsheets/factsheet.asp?fsID=98, Last Accessed January 19, 2006.

[441] United States Air Force, "USAF Fact Sheet: KC-10 Extender." Online at http://www.af.mil/factsheets/factsheet.asp?fsID=109, Last Accessed January 19, 2006.

[442] United States Air Force Research Laboratory, "Manufacturing Technology Program Reduces Maintenance Time for the B-2 Bomber Fleet." Online at http://www.ml.afrl.af.mil/stories/MLM/afrl_ws_06_0404.html. Accessed October 28, 2006.

[443] United States Air Force Research Laboratory, "Sensors Directorate Technology Thrusts." Online at http://www.afrl.af.mil/sn/thrusts.html. Last Accessed July 2006.

[444] United States Air Force Research Laboratory, "Vehicles Directorate Intranet Site." http://www.va.afrl.af.mil/. Last Accessed December 2006.

[445] United States Air Force Research Laboratory, "Trisonic Gasdynamics Wind Tunnel Facility Reopened." Online at http://www.afrl.af.mil/articles/062606_WindtunnelReopened.asp, June 2006. Last Accessed October 2006.

[446] United States Air Force Research Laboratory Vehicles Directorate, "Air Force Long Range Strike Concepts." Online at http://www.va.afrl.af.mil/IC/LRS/lrs_index.html. Accessed March 20, 2006.

[447] United States Army TRADOC DCSINT Threat Support Directorate, "World Equipment Guide." Online at http://www.fas.org/man/dod-101/sys/land/row/, September 2001.

[448] United States Department of Defense, "Defense Military Simulation Office MSRR Home Page." Online at http://www.msrr.dmso.mil/. Last Accessed October 12, 2005.

[449] United States Department of Defense, "DoD Mission Statement." Online at http://www.defenselink.mil/admin/about.html. Last Accessed August 2006.

[450] United States Department of Defense, "Report to Congress on the Conduct of the Persian Gulf War," tech. rep., 1991.

[451] United States Department of Defense, "Modeling and Simulation Master Plan." DoD 5000.59-P, October 1995.

[452] United States Department of Defense, "DoD Direction 5000.1 "The Defense Acquisition System"." Online at the Defense Acquisition Policy Center, http://akss.dau.mil/dapc/index.html, May 2003. Last Accessed January 2006.

[453] United States Department of Defense, "Force Application Functional Concept." Online at http://www.dtic.mil/futurejointwarfare/jfc.htm, March 5, 2004. Updated August 23, 2005.

[454] United States Department of Defense, "Integrated Air and Missile Defense Joint Integrating Concept Version 2.3," September 17, 2004.

[455] United States Department of Defense, "Global Strike Joint Integrating Concept, Version 1.0," January 10, 2005.

[456] United States Department of Defense, "Quadrennial Defense Review Report," February 6, 2006.

[457] United States Department of Defense, "Final Report to Congress: Conduct of the Persian Gulf War," tech. rep., April 1992.

[458] United States Department of Defense Intelligence Production Program, "Iraq Country Handbook." Online at http://www.globalsecurity.org/wmd/library/news/iraq/2002/iraq-book.htm, September 2004.

[459] United States Department of War, "Command and Employment of Air Power." Field Manual 100-20, 1943.

[460] United States Joint Chiefs of Staff, "Universal Joint Task List (UJTL)." CJCSM 3500.04C, July 1, 2002.

[461] United States Joint Chiefs of Staff, "Operation of the Joint Capabilities Integration and Development System." Chairman of the Joint Chiefs of Staff Manual, CJCMS 3170.01B, May 11, 2005.

[462] United States Missile Defense Agency, "Summary of Ballistic Missiles." Online at http://www.mda.mil/mdalink/bcmt/summary.htm. Last Accessed July 17, 2006.

[463] United States Missile Defense Agency, "Milestones of Missile Defense." Shockwave Presentation, Online at http://www.mda.mil/mdalink/shock/stones.html, Last Accessed July 17, 2006.

[464] United States Navy, "Fact File: Tomahawk Cruise Missile." http://www.chinfo.navy.mil/navpalib/factfile/missiles/wep-toma.html. Last Accessed October 20, 2005.

[465] United States Office of Naval Research, "Revolutionary Approach To Time-critical Long Range Strike (RATTLRS) Information Sheet." Online at http://www.onr.navy.mil/media/extra/fact Last Accessed December 5, 2005.

[466] University of Texas at Austin, "Perry-Castañeda Library Map Collection." Online at http://www.lib.utexas.edu/maps/iraq.html, Last Accessed April 2006.

[467] U.S. Air Force, "Fact Sheet: B-2 Spirit." http://www.af.mil/factsheets/factsheet.asp?fsID=82.

[468] U.S. Department of Defense, "DoD Dictionary." Defense Technical Information Center, Joint Publication 1-02, Online at http://www.dtic.mil/doctrine/jel/doddict, Updated April 14, 2006. Last Accessed January 2007.

[469] Vanco, M. R., "Computer Program for Design Point Performance of Turbojet and Turbofan Engine Cycles." NASA TM X-1340, NASA Lewis Research Center, Feburuary 1967.

[470] Vanderplaats, G. N., *Numerical Optimization*. Vanderplaats Research and Development, Inc., 1999.

[471] Volovoi, V., "Modeling of System Reliability Using Petri Nets with Aging Tokens," *Reliability Engineering and System Safety*, vol. 84, no. 2, pp. 149–161, 2004.

[472] Von Neumann, J., "The General and Logical Theory of Automata," in *Cerebral Mechanisms in Behavior* (L.A. Jeffress, ed.), pp. 1–32, 1951.

[473] Wallace, A. and Boldyreff, C., "Agents and Agent-Based Design Approaches to Engineering Design and Manufacturing." Presented at the Computer Aided Production Engineering (CAPE) Conference, 1999.

[474] Warden, J. A. III, *The Air Campaign: Planning for Combat*. National Defense University Press, 1988.

[475] Warden, J. A. III, *Air Theory for the Twenty-first Century*, p. 119. *Battlefield of the Future: 21st Century Warfare Issues*, Air University Press, Maxwell Air Force Base, AL, 1995.

[476] Warner Home Video, "Troy." Motion Picture, 2004.

[477] Watts, B. D., "Moving Forward on Long Range Strike," tech. rep., Center for Strategic and Budgetary Assessments, 2004.

[478] Watts, B. D., "Long Range Strike: Imperatives, Urgency, and Options," tech. rep., Center for Strategic and Budgetary Assessments, April 2005.

[479] Weatherington, D. D., "Unmanned Combat Air Systems." Powerpoint Presentation by the Office of the Undersecretary of Defense (AT&L)/PSA/Air Warfare, online at www.dtic.mil/ndia/2006psa_peo/weatherington.pdf, Last Accessed July 26, 2006.

[480] Weisstein, E., "Mathworld, The Web's Most Extensive Mathematics Resource." Online at http://mathworld.wolfram.com/, Last Accessed October 2005.

[481] Withrow, M., "AFRL Demonstrates Quantitative Technology Assesment," *news@afrl*, July 2005.

[482] Wolf, R. A., "Multiobjective Collaborative Optimization of Systems of Systems," Master's thesis, Massachusetts Institute of Technology, 2005.

[483] Wolfowitz, P., "Policy Memorandum, Cancellation of DoD 5000 Defense Acquisition Policy Documents," October 30, 2002.

[484] Wolfram, S., *A New Kind of Science*. Wolfram Media, 2002.

[485] Ye, K. Q., "Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments," *Journal of the American Statistical Association - Theory and Methods*, vol. 93, no. 444, pp. 1430–1439, December 1998.

[486] Young, R., "Modeling Conceptual Weapon Systems: FLAMES and Agent-Based Modeling." Applied Research Associates, Presented at the 2005 FLAMES User's Conference, June 2005.

[487] Young, M. J., "Agent-Based Modeling and Behavioral Representation." AFRL Horizons, Online at http://www.afrl.af.mil/techconn/index.htm, Document HE-00-09, August 2000. Last Accessed August 2006.

[488] Yu, Po-Lung, *Multiple-Criteria Decision Making, Concepts, Techniques, and Extensions*. Plenum Press, 1985.

[489] Yule, G. U., *An Introduction to the Theory of Statistics*. MacMillan Publishing Company, June 1969.

[490] Zaborowski, Michael E., "TBM Defense in the Gulf War: A Slim Margin of Victory," tech. rep., United States Army Command and Staff College, 1992.

[491] Zachman, J. A., "A Framework for Information Systems Architecture," *IBM Systems Journal*, vol. 26, no. 3, 1987.

[492] Zachman, J. A., "The Zachman Institute for Framework Advancement." Online at http://www.zifa.com/, Last Accessed March 2006.

[493] Zachman, J. A. and Sowa, J. F., "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, vol. 31, no. 3, 1992.

[494] Zaerpoor, F., Weber, R.H., "Issues in the Structure and Flow in the Pyramid of Combat Models." Presented at the 68th MORSS, US Air Force Academy, June 20-22, 2000.

[495] Zeh, J., "Net-Centric Modeling, Simulation and Analysis." Air Force Research Laboratory, Presented at the 2005 FLAMES User's Conference, June 2005.

VITA

Patrick Thomas Biltgen was born March 21, 1980 in Atlanta, GA to William and Judith Biltgen. Although he spent most of his childhood in Marietta, GA, he attended Naperville Central High School in Naperville, Illinois before enrolling in the Aerospace Engineering program at the Georgia Institute of Technology in 1998. As an undergraduate, Mr. Biltgen was a co-op student with Rolls-Royce in Indianapolis, Indiana and an undergraduate research assistant under Dr. Dimitri Mavris at the Aerospace Systems Design Laboratory (ASDL). He graduated from Georgia Tech, receiving a Bachelor of Science degree with highest honors in 2003 and a Master of Science degree in 2004. Over the past eight years, Mr. Biltgen has worked with a number of industry and government sponsors in the areas of gas turbine propulsion, missile defense, technology forecasting, strategic planning, and systems-of-systems design. He is a member of Tau Beta Pi, Sigma Gamma Tau, AIAA, INCOSE, and Kappa Kappa Psi. In 2003, he was the team leader for the first place AIAA Undergraduate Engine Design Team and also led the first place 2004 AIAA Graduate Strategic Missile Design Competition.

Mr. Biltgen is a recipient of the Tau Beta Pi Award, the Astronaut Scholarship, the Barry M. Goldwater Scholarship, the National Defense Science and Engineering Graduate Fellowship, and the 2006 AIAA Orville and Wilbur Wright Graduate Award. He was also selected as a pre-doctoral fellow in the 2005-2006 Sam Nunn Security Program under the School of International Affairs and specialized in the implications of policy on the design of future strike systems. He and his wife, Ashley, live in Atlanta, GA.